



## **PRODUCT PRICE DISPLAY USING WEMOS**

**TUGAS AKHIR**

**Program Studi  
S1 Sistem Komputer**



**Oleh:**

**DENY SETYO UTOMO  
14410200022**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM  
SURABAYA  
2018**

## ABSTRAK

Pada saat ini label harga pada bisnis ritel menggunakan kertas, label harga tersebut akan di tempel oleh karyawan. Muncul masalah jika perubahan yang dilakukan tersebut berjumlah banyak karyawan terkadang lengah dan lupa mengganti atau salah peletakan. Ini mengakibatkan kerugian pada konsumen dimana terjadi perbedaan harga antara di rak yang tertera label harga dan di kasir saat pembeli akan membayar.

Dari masalah tersebut akan di buat sebuah sistem yang dapat digunakan untuk menggantikan kertas menjadi layar LCD *display* yang dapat memberikan informasikan nama barang dan harga barang pada pembeli. Sistem yang akan dibuat akan menghubungkan antara aplikasi yang berada pada komputer dengan LCD yang berada di rak barang menggunakan jaringan *Wireless*.

Pada tugas akhir ini penulis menggunakan Wemos D1 Mini sebagai penjembaran antara data yang dikirim oleh komputer dan di tampilkan pada LCD *display* secara *Wireless*. Untuk mendistribusikan data pada LCD *display* di gunakan komunikasi I2C. Dari sistem yang telah dibuat pengiriman data nama barang dan harga barang dari komputer dapat muncul di LCD *display* secara cepat dan tidak pernah terjadi kesalahan saat pengujian.

**Kata Kunci :** Wemos D1 Mini, LCD *display*, *Wireless*, Komputer, I2C.

## DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN SYARAT .....	iii
MOTTO .....	iv
LEMBAR PERSEMBAHAN .....	v
HALAMAN PENGESAHAN.....	vi
HALAMAN PERNYATAAN .....	vii
ABSTRAK.....	viii
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xvi
DAFTAR TABEL.....	xx
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan .....	4
1.5 Sistematika Penulisan .....	4
BAB II LANDASAN TEORI.....	6
1.1 Sistem Komputer .....	6

1.2	Wemos .....	6
1.2.1	Wemos D1 Mini.....	7
1.2.2	<i>Microcontroller chipset pada microcontroller Wemos</i> .....	9
1.2.3	Fitur – fitur Wemos.....	10
1.3	Arduino IDE .....	12
1.4	I2C .....	13
1.5	<i>Liquid Crystal Display (LCD)</i> .....	15
1.6	Microsoft Visual Studio.....	16
1.7	C# .....	17
1.8	WLAN .....	18
1.9	Router .....	19
1.10	Access point .....	19
1.11	Protokol TCP/IP .....	20
1.12	Telnet .....	21
1.13	Parameter QoS .....	21
BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM .....		24
3.1	Metode Penelitian .....	24
3.2	Prosedur Penelitian .....	26
3.3	Perancangan Perangkat Keras.....	27
3.3.1	Rangkaian LCD 16x2 Ke I2C 1602 .....	28



3.3.2	Rangkaian I2C LCD Ke Wemos.....	29
3.3.3	Algoritma Penerimaan Data Dari Server .....	31
3.3.4	Algoritma Pengiriman Data Ke LCD.....	33
3.4	Perancangan Perangkat Lunak.....	34
3.4.1	Algoritma Pengiriman Data .....	35
3.5	Perancangan Topologi Jaringan.....	40
3.5.1	Configurasi IP Server.....	41
3.5.2	Configurasi Routing dan IP Router.....	43
3.5.3	Konfigurasi IP Static pada Wemos D1 mini .....	46
BAB IV HASIL PENGUJIAN DAN PENGAMATAN.....		48
4.1	Pengujian Wemos .....	48
4.1.1	Tujuan .....	49
4.1.2	Alat yang digunakan .....	49
4.1.3	Prosedur pengujian.....	49
4.1.4	Hasil Pengujian .....	54
4.2	Pengujian <i>Scanning Access point</i> oleh Wemos .....	55
4.2.1	Tujuan .....	55
4.2.2	Alat yang digunakan .....	56
4.2.3	Prosedur pengujian.....	56
4.2.4	Hasil Pengujian .....	58

4.3	Pengujian menghubungkan <i>WiFi</i> Wemos pada <i>access point</i> .....	58
4.3.1	Tujuan .....	59
4.3.2	Alat yang digunakan .....	59
4.3.3	Prosedur pengujian.....	59
4.3.4	Hasil Pengujian .....	61
4.4	Pengujian <i>Scanning</i> dan Pengiriman I2C Wemos Menuju LCD.....	62
4.4.1	Tujuan .....	63
4.4.2	Alat yang digunakan .....	63
4.4.3	Prosedur pengujian.....	63
4.4.4	Hasil Pengujian .....	65
4.5	Pengujian Jarak Jangkauan <i>WiFi</i> Wemos.....	66
4.5.1	Tujuan .....	66
4.5.2	Alat yang digunakan .....	66
4.5.3	Prosedur pengujian.....	67
4.5.4	Hasil Pengujian .....	69
4.6	Pengujian Penerimaan Data Wemos Dari PC Melalui Telnet .....	71
4.6.1	Tujuan .....	71
4.6.2	Alat yang digunakan .....	72
4.6.3	Prosedur pengujian.....	72
4.6.4	Hasil Pengujian .....	75

4.7	Pengujian Sistem .....	76
4.7.1	Tujuan .....	77
4.7.2	Alat yang digunakan .....	77
4.7.3	Prosedur pengujian.....	78
4.7.4	Hasil Pengujian .....	82
BAB V PENUTUP.....		84
5.1	Kesimpulan.....	84
5.2	Saran .....	85
DAFTAR PUSTAKA .....		86
LAMPIRAN.....		88
BIODATA PENULIS .....		98



INSTITUT BISNIS  
& INFORMATIKA  
**stikom**  
SURABAYA

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Seiring perkembangan zaman keberadaan bisnis ritel di tengah masyarakat menjadi semakin penting. Hal ini disebabkan karena adanya perubahan dalam pola berbelanja masyarakat yang semakin selektif, selain itu juga karena adanya perubahan cara pandang konsumen terhadap bisnis ritel itu sendiri. Bisnis ritel yang secara tradisional dipandang sebatas penyedia barang dan jasa telah berkembang menjadi tidak sekedar tempat berbelanja tetapi juga tempat rekreasi dan bersosialisasi, sebagai konsekuensinya bisnis ritel yang semula dikelola secara tradisional berubah menjadi bisnis yang semakin inovatif, dinamis, dan kompetitif. Menurut Suhartanto dan Nuralia, (2001,27). Perubahan gaya hidup masyarakat inilah yang melatarbelakangi hadir dan berkembangnya bisnis pasar swalayan.

Gaya hidup masyarakat ini yang mendorong semakin banyaknya swalayan baru atau cabang swalayan yang berdiri di kota-kota. Situasi ini membuat persaingan swalayan menjadi sangat kompetitif. Berbagai cara diterapkan melalui strategi pemasaran yang efektif dengan satu tujuan yaitu untuk menang dalam persaingan. Karena situasi ini berdampak kepada para pelanggan yang lebih bersikap selektif terhadap memilih swalayan untuk berbelanja.

Dalam proses penentuan dimana seseorang akan berbelanja terlihat bahwa faktor lokasi dan harga akan ikut menentukan. Keputusan berbelanja biasanya memerlukan pertimbangan yang benar-benar mendukung dan dapat menguntungkan pembeli seperti faktor harga. Harga merupakan komponen yang

berpengaruh langsung terhadap proses penentuan dimana seseorang akan berbelanja. Menurut Swastha ( 2003, 241 ).

Maka dari itu swalayan harus terus melakukan pembaruan harga agar tidak kalah dengan para pesaingnya. Adapun faktor perubahan harga barang bukan hanya dipengaruhi oleh persaingan pasar semata tetapi ada banyak faktor salah satunya adalah dipengaruhi oleh perubahan harga bahan baku barang atau harga beli barang. Hal ini menuntut swalayan harus melakukan *update* harga barang di rak-rak barang yang ada karena perubahan harga.

Saat ini label harga swalayan masih menggunakan kertas yang diprint lalu dipasang di setiap rak-rak barang oleh pegawai swalayan. Hal ini memunculkan beberapa masalah yang terjadi di swalayan seperti harga di rak berbeda dengan harga di kasir, hal ini berakibat kerugian pada pelanggan. Ini dikarenakan kelalaian pegawai dalam melakukan *update* label harga barang di rak-rak barang yang dijual.

Ada kasus terjadi di suatu minimarket di daerah Bengkulu karena lalai dalam *update* label barang, sehingga minimarket tersebut dilaporkan ke polisi, (tribunnews.com, 2017)

Kesimpulan kasus tersebut adalah perbedaan antara harga di rak dan di kasir sebesar Rp. 200,- per item, lalu indomart tersebut diduga telah melanggar Undang-Undang Nomor 8 Tahun 1999 tentang Perlindungan Konsumen. Untuk mengantisipasi hal tersebut terjadi, transmart carefour pun mengeluarkan kebijakan untuk melindungi hak yang dimiliki oleh konsumen seperti berikut “Kebijakan yang saat ini berlaku jika menemukan selisih harga antara di

rak *display* dengan di kasir adalah komitmen memberikan satu kali harga yang terendah kepada konsumen.“ menurut (finance.detik.com, 2017)

Sanksi atas kelalaian tersebut juga telah diatur oleh pemerintah melalui Permendag 35/2013 tentang Pencantuman Harga Barang Dan Tarif Jasa Yang Diperdagangkan pasal 9.

Sistem untuk harga yang digunakan saat ini hanya sebatas *update* di bagian kasir. Karena komputer pada kasir mempunyai *database* untuk harga barang, dari *database* tersebut pegawai baru bisa mencetak label barang yang berada di rak melalui sistem yang ada. Tetapi sistem ini belum sampai menjamah ke label harga di rak barang.

Seiring dengan perkembangan teknologi saat ini mencakup semua bidang dalam kehidupan manusia. Maka hal ini dapat dimanfaatkan untuk membangun sebuah sistem yang berbasis *controller* untuk menampilkan label harga dengan otomatis. Sehingga apabila *database* harga barang berubah harga di rak barang juga ikut mengalami perubahan.

*Microcontroller* merupakan sarana yang digunakan untuk membantu mewujudkan sistem label harga otomatis tersebut. *Microcontroller* tersebut akan mengendalikan LCD sebagai pengganti label harga yang selama ini menggunakan kertas. Hal ini dapat dilakukan dengan memberi pengalamatan di setiap LCD agar 1 *microcontroller* dapat memberi perintah kepada lebih dari 1 LCD.

Sehubungan dengan hal tersebut, maka penulis mengambil topik untuk menyelesaikan permasalahan di atas dari sisi teknologi. Topik tersebut adalah sistem kendali *display* harga pada rak barang dan kasir dengan judul “**PRODUCT PRICE DISPLAY USING WEMOS**”

## 1.2 Rumusan Masalah

Berdasarkan masalah yang terdapat pada latar belakang permasalahan tentang perbedaan harga di rak barang dan di kasir, maka dapat dirumuskan permasalahan sebagai berikut : Bagaimana menghindari perbedaan harga pada kasir dan rak barang pada swalayan secara mudah, cepat, dan *portable*?

## 1.3 Batasan Masalah

Pada pengimplementasian alat ini terdapat beberapa batasan masalah antara lain :

- Dalam 1 Wemos akan diberi 2 I2C LCD karakter 16x2.
- Dalam 1 *access point* akan diberi 1 *router access point*.
- Dapat mengirimkan informasi nama dan harga barang dari komputer / server ke Wemos untuk ditampilkan di LCD melalui WLAN.
- Controller* menggunakan Wemos D1 mini.
- Penelitian ini tidak membahas tentang sudut pandang bisnis dan hukum.

## 1.4 Tujuan

Tujuan dilaksanakan penelitian tentang **product price display using Wemos** sebagai inovasi teknologi untuk menjaga konsistensi harga pada rak barang dan kasir.

## 1.5 Sistematika Penulisan

### BAB I PENDAHULUAN

Dalam bab ini penulis menguraikan latar belakang pergerakan bisnis ritel , permasalahan perbedaan harga antara di rak barang dan di kasir, dan perkembangan teknologi saat ini, juga solusi yang diberikan dari segi teknologi. Perumusan Masalah, Batasan Masalah, Tujuan, dan sistematika penulisan.

sehingga penelitian dalam penyelesaian masalah tersebut memiliki titik fokus dan tidak mengambang dari judul yang telah dibuat.

## **BAB II LANDASAN TEORI**

Dalam bab ini penulis menguraikan yang terdiri dari landasan teori penelitian, berupa alat yang akan digunakan, program yang digunakan dalam penelitian pembuatan tugas akhir ini, serta teori – teori yang menunjang dalam pelaksanaan penelitian, juga referensi pengujian sistem yang dibuat pada penelitian tugas akhir ini.

## **BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM**

Dalam bab ini penulis akan menguraikan rancangan sistem yang dibuat berupa rangkaian elektronik, *hardware*, *software*, dan pemrograman. untuk menyelesaikan penelitian pembuatan tugas akhir ini.

## **BAB IV HASIL PENGUJIAN DAN PENGAMATAN**

Dalam Bab ini dibahas mengenai tahap pengujian yang berisi tujuan pengujian alat yang digunakan dan prosedur pengujian. Serta membahas hasil dari pengujian yang dilakukan.

## **BAB V PENUTUP**

Dalam bab ini penulis menyimpulkan hasil pengujian, hasil penelitian dan rancangan sistem dalam rangka menjawab tujuan penelitian yang dilakukan, serta saran-saran yang penulis berikan untuk yang berkenan melanjutkan penelitian ini.



## BAB II

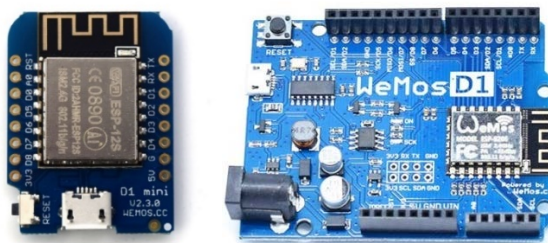
### LANDASAN TEORI

#### 1.1 Sistem Komputer

Sistem komputer adalah suatu sinergi antara perangkat lunak (*software*), perangkat keras (*hardware*) dan data, mesin pengolah data yang biasanya meliputi komputer, program aplikasi, jaringan, dan manusia untuk menghasilkan informasi. Sistem komputer ada pada hampir setiap perusahaan atau instansi untuk mendukung kegiatan bisnis mereka sehari-hari.

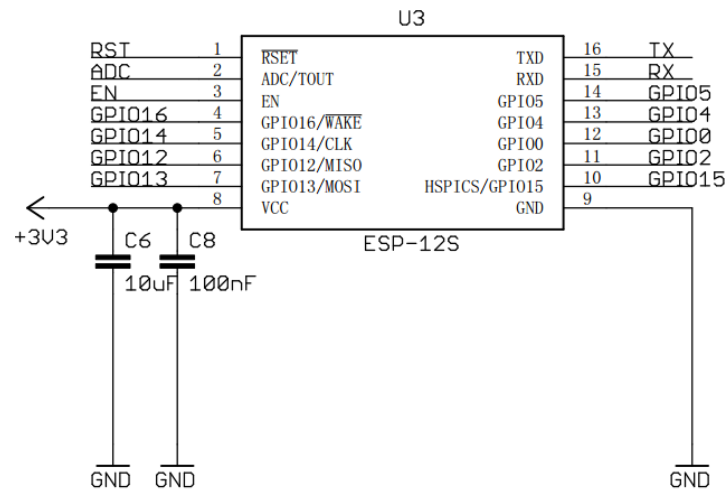
#### 1.2 Wemos

Wemos adalah sebuah *microcontroller* yang dikembangkan berbasis ESP8266. *Microcontroller* Wemos dibuat sebagai solusi dari mahalnya sebuah sistem *Wireless* berbasis *microcontroller* lainnya. Dengan menggunakan *microcontroller* Wemos biaya yang dikeluarkan untuk membangun sistem *microcontroller* berbasis *WiFi* sangat mudah dan murah, biaya yang dikeluarkan apabila membangun sistem *WiFi* dengan menggunakan *microcontroller* Arduino Uno dan *WiFi shield*. Wemos memiliki 2 versi yaitu Wemos D1 Mini dan Wemos D1 bentuk fisik dari Wemos sendiri dapat di lihat pada gambar 2.1.



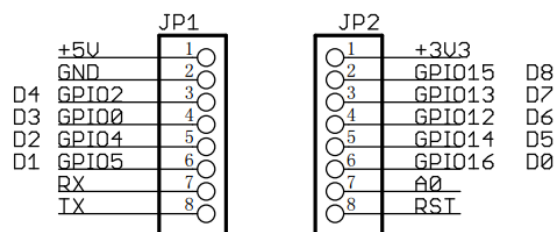
**Gambar 2.1** Board Wemos D1 Mini sebelah kiri dan Wemos D1 sebelah kanan (Sumber : D1 Mini)

### 1.2.1 Wemos D1 Mini



**Gambar 2.2** Skematik Rangkaian ESP-12S Wemos D1 Mini

Dari Gambar 2.2 dapat dilihat bentuk rangkaian ESP-12S dari Wemos D1 Mini. Wemos D1 Mini merupakan *board microcontroller open source* yang menggunakan ESP-12S sebagai *microcontrollernya*. Wemos memiliki 12 *pin input / output*, 1 *pin* digunakan untuk analog *input* maksimal 3.2V, 11 *pin* sisanya digunakan untuk digital *input / output*. Lalu juga terdapat tombol reset, koneksi USB dengan kabel data micro USB. Pada gambar 2.3 dapat dilihat perubahan pin ESP-12S menjadi inisialisasi *pin digital I/O*.



**Gambar 2.3** Pin I/O Wemos D1 Mini

Pada dasarnya pemrograman Wemos ini dapat menggunakan Arduino IDE yang sudah di *setting* pada setingan *board* Arduino IDE menjadi setingan Wemos.

Berikut adalah spesifikasi dari Wemos D1 Mini

**Tabel 2.1** Spesifikasi Wemos D1 Mini

<i>Microcontroller</i>	ESP8266EX
Operating Voltage	3.3V
Digital I/O Pins	11
Analog Input Pins	1 (Max input 3.2V)
Clock Speed	80 MHz/ 160 MHz
Flash	4M bytes
Length	34,2 mm
Width	25.6 mm
Weight	10g

Dari tabel 2.1 dapat dilihat spesifikasi Wemos D1 Mini dan Wemos dapat diaktifkan melalui koneksi USB atau dengan satu daya *external*. *External* (non – USB) daya dapat berasal dari adaptor DC atau baterai. Adaptor ini ditancapkan pada *pin* 5V pada Wemos D1 Mini.

*Board* dapat beroperasi pada rentang tegangan 3,3V – 7V. Jika tegangan kurang dari 3.3V, maka Wemos tidak akan menyala atau data akan kacau karena kurangnya konsumsi daya. Sebaliknya jika tegangan lebih dari 5V, maka Wemos akan terbakar dan rusak karena kelebihan tegangan.

*Pin* listrik pada Wemos D1 Mini :

5V : Dapat digunakan sebagai *input* atau *output* sumber daya sebesar 5V. Jika daya *external* bukan dari USB, maka *pin* ini untuk masukan daya, bila daya sudah menggunakan USB maka *pin* ini sebagai *output* tegangan.

3,3V : Dapat digunakan sebagai *input* atau *output* sumber daya 3,3V, jika daya *external* bukan dari USB maka *pin* ini untuk masukan

daya, bila daya sudah menggunakan USB maka *pin* ini sebagai *output* tegangan.

GND : *Pin* untuk penetral *noise* atau juga berperan sebagai 0V, pada aplikasi ini.

Pada Wemos ini memiliki *Clock speed* 160Mhz, konektivitas *WiFi*, dan memori yang digunakan cukup besar yaitu 4 MB. Dalam operasi kerjanya Wemos ini dapat bekerja direntang suhu antara 40 °C - 125 °C. Walaupun modul ini bukan Arduino, namun modul ini mendukung pemrograman menggunakan Arduino IDE, beserta *library* dan fungsi-fungsinya yang lain. Untuk koneksi ke *laptop* atau ke satu daya *microcontroller* Wemos ini menggunakan konektor *micro USB* yang umum digunakan untuk kabel data *smartphone* Android.

### 1.2.2 *Microcontroller chipset* pada *microcontroller* Wemos

Pada *microcontroller* Wemos memiliki 2 buah *chipset* yang digunakan sebagai otak kerja perangkat tersebut antara lain adalah :

#### a. *Chipset* ESP8266

*Chipset* ESP8266 adalah sebuah *chip microcontroller* yang memiliki fitur *WiFi* yang mendukung *stack* TCP/IP. *Chip* ini memungkinkan *microcontroller* untuk terhubung ke jaringan *WiFi* pada frekuensi 2.4GHz dan membuat koneksi TCP/IP hanya dengan *command* yang sederhana seperti gaya *hayes*. Dengan *clock* 80 MHz *chip* ini dibekali dengan 4MB *external* RAM, mendukung format IEEE 802.11 b/g/n sehingga tidak menyebabkan interferensi bagi yang lain. Dari sisi keamanan *chip* ini sudah cukup aman digunakan karena mendukung

enkripsi WEP dan WPA. *Chip* ini mempunyai 16 GPIO *pin* yang bekerja pada 3.3 Volt, dan 1 *pin* ADC dengan resolusi 10 *bit*.

#### **b. Chipset CH340**

*Chipset* CH340 adalah sebuah *chip* yang berfungsi untuk mengubah USB menjadi *serial interface*. Sebagai contohnya adalah aplikasi USB converter IrDA atau USB *converter to printer*. Dalam mode *serial interface*, *chip* ini digunakan untuk memperbesar sinyal *asynchronous serial interface* komputer atau mengubah perangkat *serial interface* umum untuk berhubungan dengan *bus* USB langsung.

#### **1.2.3 Fitur – fitur Wemos**

Ada beberapa fitur-fitur perangkat keras yang dapat ditemukan pada modul *microcontroller* Wemos antara lain sebagai berikut :

##### **a. Pin Digital**

*Microcontroller* Wemos secara fisik mempunyai *pin* digital berjumlah 9 *pin* yang dimulai dari d0 – d8. Namun secara program mempunyai 16 *pin* digital yang beberapa *pin* didefinisikan menjadi 2 alamat I/O. *Pin* digital ini dapat digunakan menjadi *input* maupun *output* sama fungsinya dengan *pin* digital *input output* pada Arduino maupun *microcontroller* yang lain. Selain itu *pin* digital pada *microcontroller* Wemos sudah dapat digunakan untuk PWM (*Pulse Width Modulator*)

##### **b. Pin analog**

*Microcontroller* Wemos hanya mempunyai 1 buah *pin* analog yang dapat digunakan sebagai *input* untuk ADC yang memiliki 10 *bit* resolusi dengan nilai tegangan maksimal 3.2 Volt. *Pin* ini juga dapat digunakan

sebagai *pin* digital *input output*. Selain itu *pin* ini juga memiliki resistor *pullup* namun untuk menggunakan *pullup* ini cukup membuat repot karena ada beberapa aturan yang harus dilakukan terlebih dahulu.

**c. Memori**

Ada 3 jenis memori yang digunakan dalam *microcontroller* Wemos ini, antara lain :

1. RAM untuk menyimpan memori instruksi sebesar 64KB
2. RAM untuk menyimpan data sebesar 96KB
3. *External QSPI flash* untuk menimpa listing program sebesar 4MB

**d. I2C**

*Microcontroller* Wemos ini didukung dengan I2C yang berada pada D4 sebagai *serial data* dan D5 sebagai *serial clock*. Yang akan lebih memudahkan untuk perangkat ini mengendalikan *hardware* lain yang mendukung I2C pula.



INSTITUT BISNIS  
& INFORMATIKA  
**stikom**  
SURABAYA

**Tabel 2.2** Konfigurasi *Pin* Wemos

Pin	Fungsi	ESP-8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	Analog input, max 3.3 input	A0
D0	I/O	GPIO16
D1	I/O, SCL	GPIO5
D2	I/O, SDA	GPIO4
D3	I/O, 10k Pull-up	GPIO0
D4	I/O, 10k Pull-up, Builtin_Led	GPIO2
D5	I/O, SCK	GPIO14
D6	I/O, Mosi	GPIO12
D7	I/O, Miso	GPIO13
D8	I/O, 10k Pull-Down, SS	GPIO15
G	Ground	GNF
5V	5V	-
3V3	3,3V	3,3V
RST	Reset	Reset

Dari tabel 2.2 diketahui fungsi dari masing-masing pin pada Wemos D1 Mini beserta fungsinya seperti dapat dilihat pada rangkaiannya pada gambar 2.3.

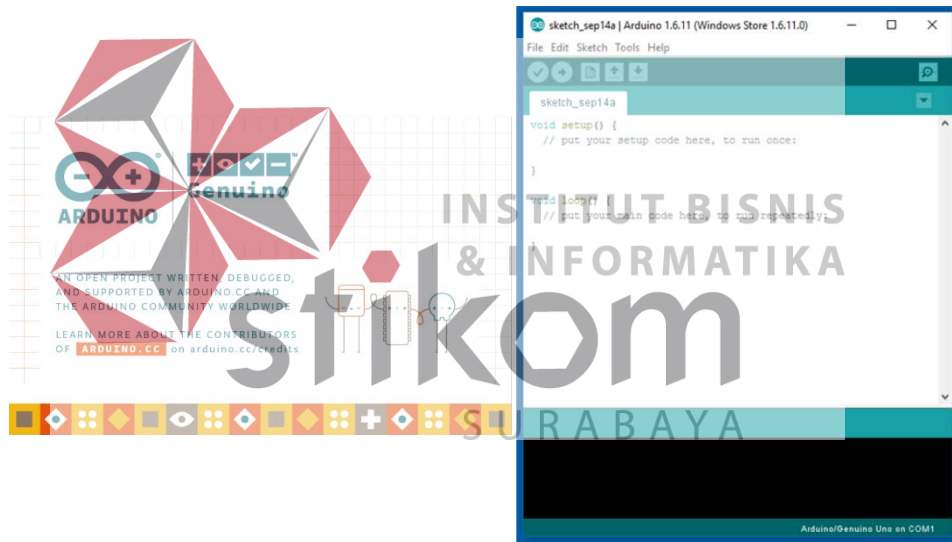
### 1.3 Arduino IDE

*Software* Arduino IDE adalah *software* yang digunakan untuk membuat program untuk memberi perintah kepada Arduino. Dengan menggunakan bahasa C++ yang dikembangkan oleh Arduino. Pemrograman dengan Arduino IDE dimudahkan lagi dengan banyaknya *library* yang disediakan. *Software* IDE Arduino terdiri dari 3 (tiga) bagian antara lain :

1. *Editor program*, untuk menulis dan mengedit program dalam bahasa *processing*. *Listing* program pada Arduino disebut *sketch*.
2. *Compiler*, sebuah modul yang berfungsi mengubah bahasa *processing* (kode program) menjadi kode biner karena kode biner merupakan bahasa program yang dipahami oleh *microcontroller*.

3. *Uploader*, sebuah modul yang berfungsi memasukkan kode biner ke dalam memori *microcontroller*.

Perangkat lunak yang ditulis menggunakan Arduino IDE disebut *sketch*. *Sketch* ditulis pada editor teks dan disimpan dengan *extensifile* .ino. Struktur perintah pada Arduino secara garis besar terdiri dari dua bagian yaitu *void setup* dan *void loop*. *Void setup* berisi perintah yang akan dieksekusi hanya satu kali sejak Arduino dihidupkan, sedangkan *void loop* berisi perintah yang akan dieksekusi berulang-ulang selama Arduino dinyalakan. Untuk tampilan software arduino IDE dapat dilihat pada gambar 2.4.



**Gambar 2.4** Tampilan Arduino IDE dan *Sketch*

#### 1.4 I2C

I2C singkatan dari *Inter Integrated Circuit*, adalah sebuah *protocol* yang digunakan pada *multi-master serial computer bus* yang diciptakan oleh Philips yang digunakan untuk saling berkomunikasi dengan perangkat *low-speed* lainnya yang diaplikasikan pada *motherboard*, *embedded system*, atau *cellphone* menurut (Frans, 2007). untuk komunikasi *serial* antar IC, dan sering disebut juga *Two Wire*



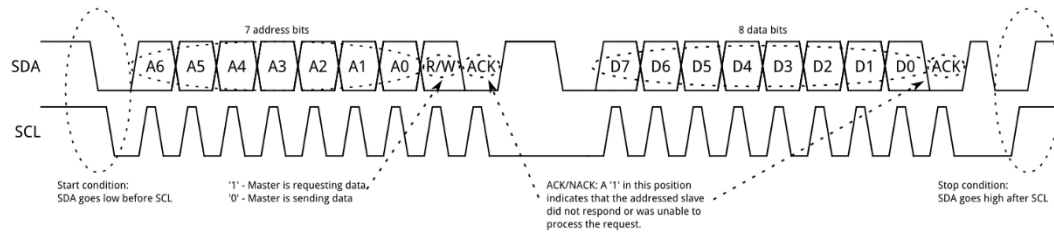
*Interface* (TWI). *Bus* yang digunakan untuk komunikasi antara *mikrokontroler* dan perangkat seperti LCD, memori, sensor, temperatur, *I/O expander* dan lain sebagainya. Dalam hal ini I2C akan dimanfaatkan untuk mengkomunikasikan antara *microcontroller* dengan LCD karakter 16x2 menggunakan I2C 1602. Bentuk fisik dari I2C 1602 sendiri dapat dilihat pada gambar 2.5.



**Gambar 2.5** Modul I2C 1602 (Sumber : Panduan lengkap penggunaan 1602 / 2004 I2C LCD untuk arduino)

Komunikasi I2C dilakukan melalui dua jalur: SDA (*serial data*) dan SCL (*serial clock*). Setiap perangkat I2C memiliki 7-bit alamat yang unik. MSB (*Most Significant Bit*) adalah *fix* dan ditujukan untuk kategori perangkat. Sebagai contoh, 1010 biner ditujukan untuk *serial* EEPROM. Tiga *bit* berikutnya memungkinkan 8 kombinasi alamat I2C, yang berarti, dimungkinkan 8 perangkat dengan tipe yang sama, beroperasi pada bus I2C yang sama. Pengiriman data hanya dapat dimulai ketika saluran tidak sibuk, ditandai dengan kondisi *high* yang cukup lama pada *pin* SCL maupun SDA seperti ditunjukkan pada Gambar

Selama pengiriman data, saluran data (SDA) harus dalam keadaan *stabil* ketika saluran *clock* (SCL) dalam keadaan *high*. Perubahan kondisi SDA pada saat SCL *high* akan dianggap sebagai sinyal-sinyal kendali, seperti: sinyal mulai (*high* ke *low*) atau sinyal berhenti (*low* ke *high*). Gambaran pengiriman data I2C dapat dilihat pada gambar 2.6.



**Gambar 2.6** Sinyal Data I2C (Sumber : Pengetahuan Dasar pemrograman C untuk I2C AVR dengan codevision)

### 1.5 *Liquid Crystal Display (LCD)*

*Display* elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf maupun grafik. LCD adalah suatu jenis *display* elektronik. LCD merupakan media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan diberbagai bidang misalnya alal-alat elektronik seperti televisi, kalkulator dan layar komputer. Salah satu jenis LCD (*Liquid Crystal Display*) yang sering digunakan ialah LCD *dot* matrik dengan jumlah karakter 16x2. Gambar LCD seperti yang ditunjukkan pada Gambar 2.6. Karakteristik yang dimiliki LCD ini adalah sebagai berikut:

- Terdiri dari 16 karakter dan 2 baris.
- Mempunyai 192 karakter tersimpan.
- Terdapat karakter generator terprogram.
- Dapat dialamati dengan mode 4-bit dan 8-bit.
- Dilengkapi dengan *back light*.



**Gambar 2.7** LCD Karakter 16x2 (Sumber : Panduan lengkap penggunaan 1602 / 2004 I2C LCD untuk arduino)

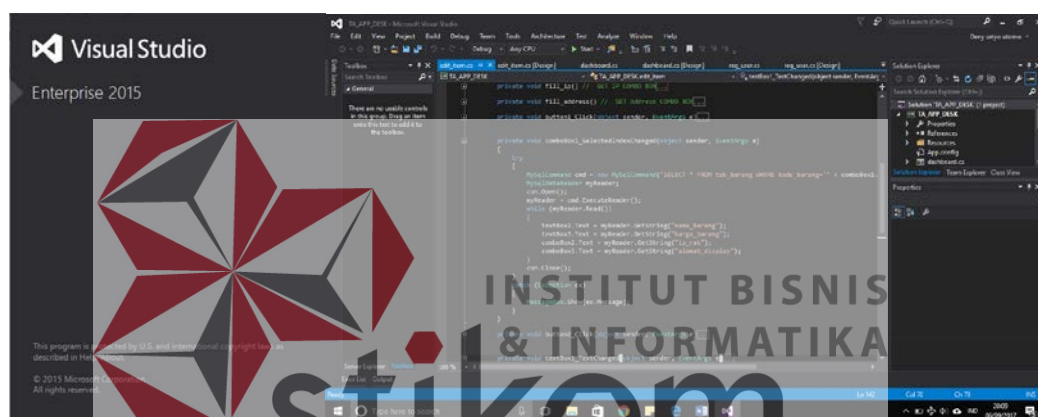
LCD sebagai salah satu jenis *display* elektronik bekerja dengan tidak menghasilkan cahaya. LCD memantulkan cahaya yang ada disekitarnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. Spesifikasi kaki LCD 16x2 yaitu:

1. *Pin 1* merupakan *Ground*.
2. *Pin 2* merupakan *VCC*.
3. *Pin 3* merupakan pengatur kontras.
4. *Pin 4* merupakan *RS (Register Select) instruction*.
5. *Pin 5* merupakan *R/W (Read/Write) LCD register*.
6. *Pin 6* merupakan *EN (Enable)*.
7. *Pin 7-14* merupakan *I/O (Input/Output) data*.
8. *Pin 15* merupakan *VCC*.
9. *Pin 16* merupakan *Ground*.

#### 1.6 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *web*. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual Source Safe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan diatas Windows) ataupun *managed code* (dalam bentuk Microsoft diatas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan diatas .NET Compact Framework). Pada gambar 2.8 dapat diketahui tampilan start awal saat Visual Studio dibuka pada sebelah kiri dan tampilan untuk menulis program pada sebelah kanan.



Gambar 2.8 Tampilan dan *User Interface* Visual Studio

## 1.7 C#

Microsoft Visual C# adalah sebuah program alat bantu pemrograman (*Rapid Application Development tool*) yang dibuat oleh Microsoft Corporation dan dapat digunakan untuk membuat program berbasis grafis dengan menggunakan bahasa pemrograman mirip C++. Program ini telah dimasukkan ke dalam produk Microsoft Visual Studio, bersama-sama dengan Visual C++, Visual Basic, Visual FoxPro serta Visual J#. Se jauh ini, program ini merupakan program yang paling banyak digunakan oleh para programmer untuk membuat program dalam bahasa C#.

Microsoft Visual C# adalah sebuah bahasa pemrograman yang bersifat OOP (*Object Oriented Programming*) yang dikembangkan oleh Microsoft Corp guna menggantikan visual J# yang terhalang lisensi dengan pemilik java pada saat itu Sun Microsystem, yang akhirnya membuat Microsoft untuk merancang sebuah bahasa pemrograman baru yang lebih dinamis

## 1.8 WLAN

WLAN atau *Wireless Local Area Network* merupakan sekumpulan komputer yang saling terhubung antara satu dengan lainnya sehingga terbentuk sebuah jaringan komputer dengan menggunakan media udara/gelombang sebagai jalur lintas datanya. Pada dasarnya *Wireless* dengan LAN merupakan sama-sama jaringan komputer yang saling terhubung antara satu dengan lainnya, yang membedakan antara keduanya adalah media jalur lintas data yang digunakan, jika LAN masih menggunakan kabel sebagai media lintas data, sedangkan *Wireless* menggunakan media gelombang radio/udara.

WLAN sendiri adalah kependekan dari *Wireless Local Area Network* terdiri dari dua jenis model yaitu model *AdHoc* dan model infrastruktur. Sebenarnya jaringan *Wireless* LAN hampir sama dengan jaringan LAN Kabel, akan tetapi setiap node pada WLAN menggunakan piranti *Wireless* agar dapat berhubungan dengan jaringan, node pada WLAN menggunakan kanal frekuensi yang sama dan SSID yang menunjukkan identitas piranti *Wireless*. Yang membedakan jaringan kabel LAN, jaringan *Wireless* memiliki dua model yang dapat digunakan yaitu infrastruktur dan *Ad-Hoc*. Konfigurasi infrastruktur adalah komunikasi antar masing-masing PC melalui sebuah *access point* pada WLAN

atau LAN. Pada gambar 2.9 dapat dilihat gambar simulasi *wireless* yang terhubung pada perangkat.



**Gambar 2.9** Koneksi *WiFi* ke Perangkat

### 1.9 *Router*

*Router* adalah perangkat keras jaringan komputer yang dapat digunakan untuk menghubungkan beberapa jaringan yang sama atau berbeda. *Router* adalah sebuah alat untuk mengirimkan paket data melalui jaringan atau internet untuk dapat menuju tujuannya, proses tersebut dinamakan *routing*.

*Router* memiliki fungsi utama untuk membagikan atau mendistribusikan *IP Address*, baik itu secara statis ataupun DHCP (*Dynamic Host Configuration Protocol*) kepada semua komputer yang terhubung ke *router* tersebut. Dengan adanya *IP Address* unik yang dibagikan *router* tersebut kepada setiap komputer dapat memungkinkan untuk saling terhubung serta melakukan komunikasi, baik itu pada LAN, WLAN, atau internet.

### 1.10 *Access point*

*Access point* adalah sebuah perangkat jaringan yang berisi sebuah *transceiver* dan antena untuk transmisi dan menerima sinyal ke dan dari *clients remote*. Dengan *access point* (AP) *clients Wireless* bisa dengan cepat dan mudah

untuk terhubung ke jaringan LAN kabel secara *Wireless*. Pada gambar 2.10 dapat dilihat contoh gambar fisik *access point*



**Gambar 2.10** *Access point* (Sumber : Tutorial Cara Setting Mikrotik RB750 via Winbox)

### 1.11 Protokol TCP/IP

*TCP/IP* (*Transport Control Protocol/Internet Protocol*) adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data pada *Wide Area Network* (WAN). *TCP/IP* terdiri dari sekumpulan protokol yang masing-masing bertanggungjawab atas bagian-bagian tertentu dari komunikasi data. Berkat prinsip ini, tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain, sepanjang ia masih bisa saling mengirim dan menerima data.

Berkat penggunaan prinsip ini, *TCP/IP* menjadi protokol komunikasi data yang fleksibel. *TCP/IP* dapat diterapkan dengan mudah di setiap jenis komputer dan *interface* jaringan, karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Agar *TCP/IP* dapat berjalan di atas *interface* jaringan tertentu, hanya perlu dilakukan perubahan pada protokol yang berhubungan dengan *interface* jaringan saja.

### 1.12 Telnet

Telnet adalah sebuah protokol jaringan yang digunakan pada internet atau *Local Area Network* untuk menyediakan fasilitas komunikasi berbasis teks interaksi dua arah yang menggunakan koneksi *virtual terminal*. Pada dasarnya fungsi Telnet adalah untuk dapat mengakses komputer dari jarak jauh. Karena Telnet dapat memungkinkan komputer penggunanya menjadi terminal dari komputer yang lain di jaringan komputer. Dan Telnet memungkinkan penggunanya dapat melakukan pengiriman data berupa teks.

### 1.13 Parameter QoS

QoS (*Quality of Service*) adalah teknologi yang diterapkan pada jaringan WAN (*Wide Area Network*) yang memungkinkan administrator jaringan untuk dapat menangani berbagai efek akibat terjadinya kemacetan (*congestion*) pada lalu lintas aliran paket di dalam jaringan. Parameter QoS juga dapat digunakan untuk mengukur efektivitas suatu *traffic network* dengan beberapa parameter besaran teknis, yaitu :

- *Delay* adalah waktu tunda yang disebabkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. *Delay* dapat dipengaruhi oleh jarak antara pengirim dan penerima, juga karena lalu lintas pada jaringan penuh yang mengakibatkan *delay* cukup besar. Berikut table parameter kategori *delay* berdasar lama *delay*nya.



Tabel 2.3 **Kategori Delay**

Kategori Delay	Besar Delay
Sangat Bagus	<150 ms
Bagus	151 s/d 300 ms
Sedang	301 s/d 450 ms
Jelek	>451 ms

Untuk mengukur nilai *delay* digunakan persamaan sebagai berikut :

$$\text{Delay rata - rata} = \frac{\text{Total delay transmisi}}{\text{Total paket yang diterima}} \dots\dots\dots (2.1)$$

- *Throughput* jaringan adalah tingkat rata-rata keberhasilan pengiriman pesan melalui saluran komunikasi. Data ini dapat disampaikan melalui link fisik atau logis, atau melewati simpul jaringan tertentu. *Throughput* biasanya diukur dalam *bit* per detik (*bit / s* atau bps), dan kadang-kadang dalam paket data per detik atau data paket per slot waktu.

Tabel 2.4 **Kategori Throughput**

Kategori Throughput	Throughput
Sangat Bagus	> 750 bps
Bagus	750 bps – 501 bps
Sedang	500 bps – 101 bps
Jelek	< 100 bps

Untuk mengukur nilai *throughput* menggunakan persamaan sebagai berikut:

$$\text{Throughput} = \frac{\text{Paket data yang diterima}}{\text{Lama waktu pengambilan paket}} \dots\dots\dots (2.2)$$

- Paket *Loss* adalah perbandingan seluruh paket IP yang hilang dengan seluruh paket IP yang dikirimkan antara pada *source* dan *destination*.

Salah satu penyebab paket *loss* adalah antrian yang melebihi kapasitas *buffer* pada setiap *node*.

Tabel 2.5 **Kategori *Packet Loss***

Kategori Degradasi	Packet Loss %
Sangat Bagus	0
Bagus	3
Sedang	15
Jelek	25

Untuk mengukur nilai *Packet Loss* menggunakan persamaan sebagai berikut:

$$\text{Packet Loss} = \frac{\text{Data yang dikirim} - \text{Paket data yang diterima}}{\text{Packet data yang dikirim}} \dots (2.1)$$



## BAB III

### METODE PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian



Gambar 3.1 Blok Diagram Rancangan Penelitian

Metode penelitian yang digunakan dalam pengerjaan tugas akhir ini adalah rancang bangun yang merupakan rancang bangun *hardware* berupa Wemos untuk penerimaan data dari server lalu menampilkan data tersebut ke LCD karakter melalui jalur komunikasi I2C. Pada rancang bangun *software* berupa membuat sebuah aplikasi untuk *input* data dari *user* dan mengirimkan data tersebut ke Wemos melalui jaringan WLAN. Dengan ini penulis berusaha untuk menguji *WiFi* yang dimiliki Wemos untuk menerima suatu informasi data dari jaringan dan mengolah informasi sesuai dengan permasalahan yang terjadi. Studi kepustakaan dilakukan untuk mencari teori atau informasi dari buku, materi

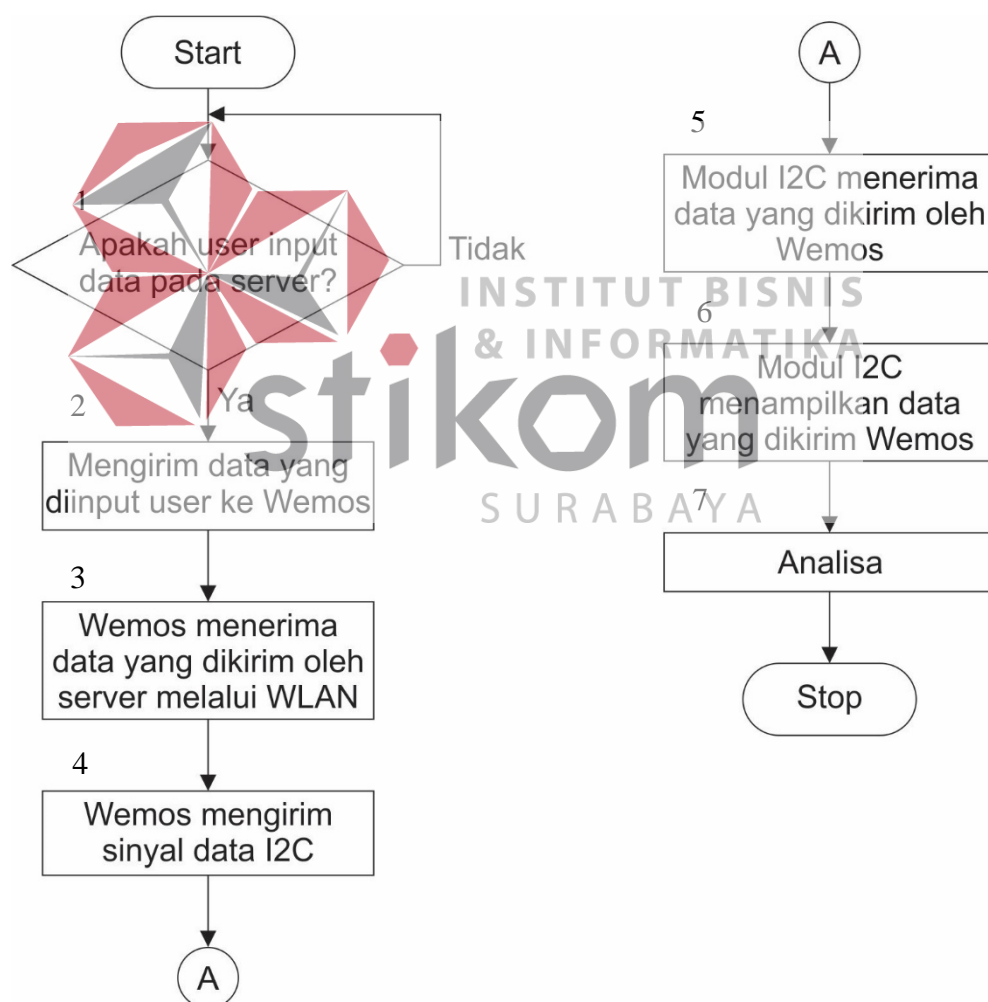
perkuliahaan serta literatur dari internet, jurnal, dan artikel – artikel yang berkaitan dengan penelitian.

Penjelasan blok diagram pada rancangan penelitian yang menjelaskan tentang proses pengiriman data dari server menuju Wemos.

1. Pada proses awal *user* akan melakukan *input* data ke server. Data yang akan dimasukkan oleh *user* adalah berupa teks IP tujuan, alamat I2C, nama barang dan harga barang yang ingin dimunculkan pada LCD karakter. Dimana IP tujuan adalah Wemos yang dituju, alamat tujuan adalah alamat I2C yang dimiliki oleh LCD. IP dan alamat ini harus diberikan agar data yang dikirimkan tepat sampai tujuan.
2. Pada bagian server akan mengolah *input* data yang diberikan oleh *user* data tersebut akan dikirimkan kepada Wemos. Data yang dikirimkan oleh server adalah berupa alamat I2C, nama barang, dan harga barang. Server akan mengirimkan data tersebut kepada Wemos yang memiliki IP yang telah dimasukkan oleh *user*.
3. Paket data yang dikirimkan oleh server pada *router* akan dilihat IP nya. Lalu *router* akan menyebarkan *broadcast* kepada semua yang terhubung ke *Accses Point*. *Router* ini berfungsi untuk membuat jalan paket data yang diterima dan mengirimkan kepada IP tujuan.
4. Paket data yang dikirimkan akan ditangkap oleh Wemos yang memiliki IP sesuai. Wemos akan membuka paket data yang berisi alamat I2C, nama barang, dan harga barang. Wemos akan mengambil data alamat I2C untuk mengirimkan data lagi ke LCD karakter melalui module I2C yang berupa nama barang dan harga barang.

5. Data yang dikirimkan oleh Wemos melalui jalur I2C (SDA dan SCL) akan ditangkap oleh modul I2C. Modul I2C ini berfungsi menangkap data yang dikirimkan oleh Wemos sesuai dengan alamat yang dimilikinya, lalu menerjemahkan data I2C tersebut menjadi tampilan di LCD karakter.
6. Melakukan pengujian dengan parameter *throughput*, paket *loss*, dan *delay* mulai dari paket dikirimkan dari server sampai ditampilkan ke LCD karakter.

### 3.2 Prosedur Penelitian



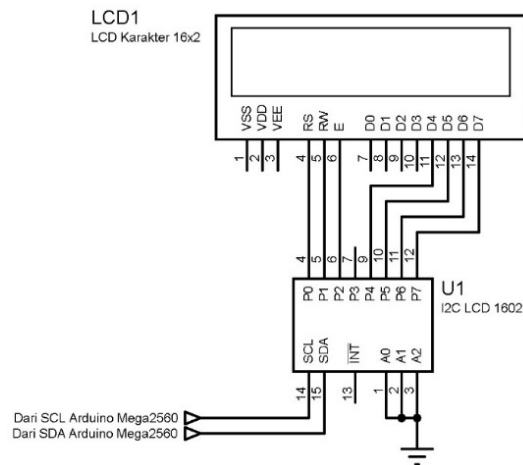
**Gambar 3.2** *Flowchart* Proses Penelitian

1. Proses dimulai dengan *user* melakukan *input* data di komputer server. Selanjutnya data yang dimasukkan oleh *user* akan dikirimkan oleh server menuju ke Wemos melalui jaringan.
2. Setelah Wemos menerima data dari server selanjutnya Wemos akan menerjemahkan data yang diterima dari server. Lalu setelah data diterjemahkan dan sesuai Wemos akan mengeluarkan sinyal I2C melalui *pin* out SDA dan SCL
3. Data I2C yang dikeluarkan oleh Wemos akan diterima oleh modul I2C yang mempunyai alamat sesuai.
4. Lalu modul I2C akan menerjemahkan data yang diterima dari Wemos untuk ditampilkan di layar LCD. Lalu analisa yang akan dilakukan adalah apakah data yang dikirimkan oleh server sampai pada LCD karakter yang benar juga data yang dikirimkan benar, tidak ada yang berubah atau hilang.

### 3.3 Perancangan Perangkat Keras

Perangkat keras adalah perangkat fisik yang berupa komponen elektronik yang mempunyai fungsi tertentu. Pada sistem ini perangkat keras dibagi menjadi 3 bagian, yaitu bagian sistem pengelola informasi atau *database*, bagian sistem jaringan dan komunikasi, dan bagian sistem elektronik untuk penampilan (*display*).

### 3.3.1 Rangkaian LCD 16x2 Ke I2C 1602



**Gambar 3.3** Rangkaian I2C ke LCD Karakter 16x2

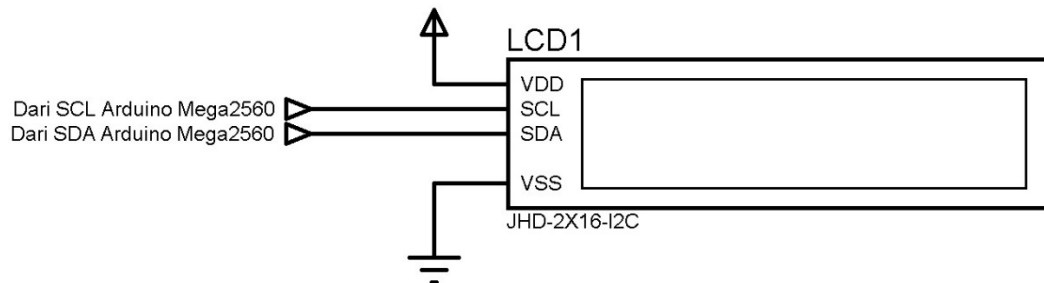
Pada Gambar 3.3 dapat dilihat bahwa *pin out* P0 – P7 menjadi *output* dari I2C yang sudah diterjemahkan oleh modul I2C 1602. Ini karena *pin* P0 – P7 menuju ke LCD Karakter 16x2 dimana LCD pada sistem ini adalah *end device* sistem yang menampilkan nama barang dan harga barang.

Sedangkan *pin out* A0 – A2 adalah alamat dari I2C 1602. Dilihat dari tabel 3.1 agar alamat I2C diatur pada I2C 1602.

**Tabel 3.1** Alamat I2C 1602

INPUTS			Address I2C
A2	A1	A0	
L	L	L	20 (hexadecimal)
L	L	H	21 (hexadecimal)
L	H	L	22 (hexadecimal)
L	H	H	23 (hexadecimal)
H	L	L	24 (hexadecimal)
H	L	H	25 (hexadecimal)
H	H	L	26 (hexadecimal)
H	H	H	27 (hexadecimal)

Bila 2 perangkat tersebut disambungkan maka *pin* keluaran dari perangkat tersebut hanya ada 4 buah yaitu terdiri dari SDA, SCL, VCC, dan GND. Dapat dilihat pada gambar 3.4.



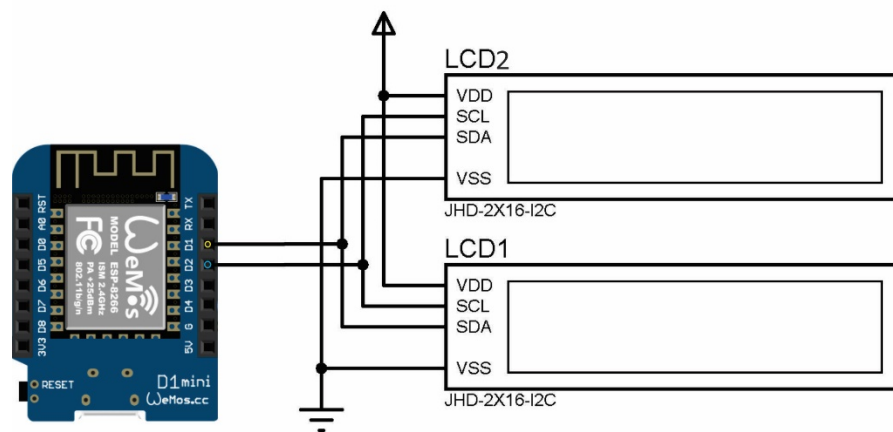
**Gambar 3.4** Gabungan Antara I2C dan LCD Karakter 16x2

### 3.3.2 Rangkaian I2C LCD Ke Wemos

I2C LCD mempunyai 4 *pin out* yaitu VDD, SCL, SDA, dan VSS. Setiap *pin* mempunyai fungsi masing seperti VDD dan VSS sebagai *pin* untuk sumber daya VDD digunakan untuk +5V dan VSS digunakan untuk GND atau 0V.

Wemos mempunyai 9 digital *pin out* dimana yang akan digunakan adalah *pin* yang mendukung komunikasi I2C yaitu *pin* D1 dan D2. *Pin* D2 adalah sebagai *serial data* (SDA) ini digunakan untuk jalur data keluar masuk dari atau ke Wemos sedangkan *pin* D1 adalah sebagai *serial clock* (SCL) ini adalah sebagai jalur *clock* yang digunakan untuk pembacaan data.



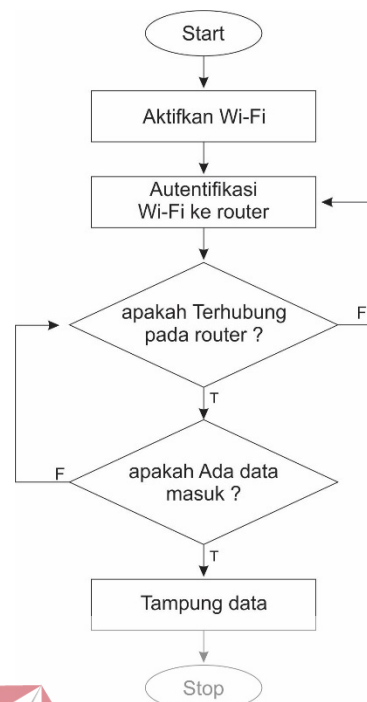


**Gambar 3.5** Rangkaian I2C LCD Karakter

Pada Gambar 3.5 dapat diketahui bahwa *pin out* Wemos yang mendukung I2C berada di *pin* D1 dan D2. Dimana *serial data* (SDA) berada pada *pin* 20 dan *serial clock* (SCL) berada di *pin* 21.

Jalur I2C dari Wemos lalu disambungkan secara paralel ke jalur I2C LCD. Pada Gambar di atas penulis menggunakan 4 LCD yang akan digunakan sebagai contoh. Jalur I2C ini dapat dirangkai secara paralel karena Wemos akan memberikan alamat pada setiap data yang dikirimkan. Apabila alamat yang dikirimkan Wemos dalam paket data cocok dengan alamat yang dimiliki I2C LCD, maka data yang dikirimkan Wemos akan diterima dan ditampilkan pada LCD.

### 3.3.3 Algoritma Penerimaan Data Dari Server



Gambar 3.6 Algoritma Penerimaan Data Dari Server

Pada saat Wemos menerima data digunakan algoritma seperti Gambar 3.6.

Awal mula *WiFi* yang dimiliki oleh Wemos dinyalakan, dimana default dari Wemos sendiri adalah *auto* aktif saat dinyalakan. Lalu Wemos akan melakukan *autentifikasi* dengan router yang SSID-nya dituju. Sebelum melakukan hal tersebut penulis memasukan *header* dan variabel yang digunakan untuk menyimpan *ssid* dan *password* untuk melakukan *autentifikasi*.

```

#include <ESP8266WiFi.h>
const char* ssid = "*****";
const char* password = "*****";

```

Setelah *header* dan *ssid* masuk, juga *password* sudah diisi, maka Wemos siap dihubungkan ke router dengan perintah `WiFi.begin(ssid,pass)` Wemos akan memulai mengautentifikasi. Seperti pada *source code* berikut :

```
void setup()
{
    WiFi.begin(ssid, password);
}
```

Untuk memastikan apakah Wemos sudah terhubung atau belum, kita membutuhkan sinyal indikator untuk mengetahui status tersebut fungsi perintah yang penulis gunakan adalah dengan `WiFi.status() == WL_CONNECTED`, perintah tersebut adalah untuk melakukan pengecekan kondisi apakah *WiFi* sudah terhubung, bila status benar maka lampu indikator pada Wemos akan berkedip.

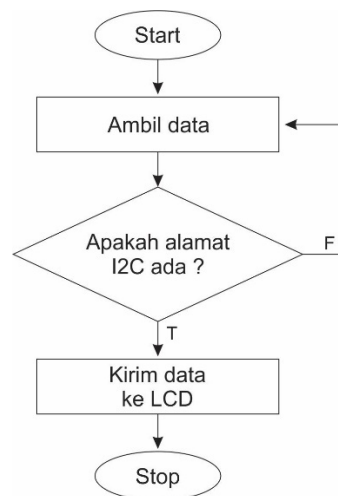
Perhatikan *source code* berikut :

```
IF (WiFi.status() == WL_CONNECTED)
{
    digitalWrite(LED_BUILTIN, LOW);
    delay(500);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(500);
}
```

Untuk *source code* lengkapnya perhatikan di bawah ini :

```
#include <ESP8266WiFi.h>
const char* ssid = "*****";
const char* password = "*****";
void setup()
{
    WiFi.begin(ssid, password);
}
void loop()
{
    IF (WiFi.status() == WL_CONNECTED)
    {
        digitalWrite(LED_BUILTIN, LOW);
        delay(500);
        digitalWrite(LED_BUILTIN, HIGH);
        delay(500);
    }
}
```

### 3.3.4 Algoritma Pengiriman Data Ke LCD



**Gambar 3.7** Algoritma Pengiriman Data Ke LCD

Pada saat Wemos menampilkan data pada LCD karakter 16x2 digunakan algoritma seperti Gambar 3.7. Awal mula pada program diberi *header* yang berfungsi untuk menjalankan I2C yang pertama adalah *wire.h* ini adalah *header* untuk membuka komunikasi I2C dan *header LiquidCristal I2C* ini adalah *header* yang digunakan untuk mengkonversi perintah *print* untuk LCD tanpa I2C menjadi data I2C untuk mengendalikan modul I2C 1602. Berikut *source code* yang akan digunakan :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Setelah *header* dimasukkan langkah yang penulis ambil selanjutnya adalah instalasi variabel untuk I2C dalam instalasi variabel ini diikuti oleh alamat I2C, banyak karakter horisontal yang dimiliki LCD dan banyak karakter vertikal yang dimiliki LCD pada penelitian ini, penulis menggunakan LCD karakter 16x2. Dengan *source code* berikut :

```
LiquidCrystal_I2C LCD1(0x3E,16,2);
```

Variabel LCD1 untuk I2C tidak dapat digunakan sebelum diberikan perintah mulai langkah selanjutnya yang harus dilakukan adalah menjalankan perintah mulai untuk I2C tersebut, dalam penulisannya perintah mulai diikuti oleh alamat I2C, banyak karakter horisontal yang dimiliki LCD dan banyak karakter vertikal yang dimiliki LCD. Perhatikan *source code* di bawah :

```
LCD1.begin(0x3E,16,2);
```

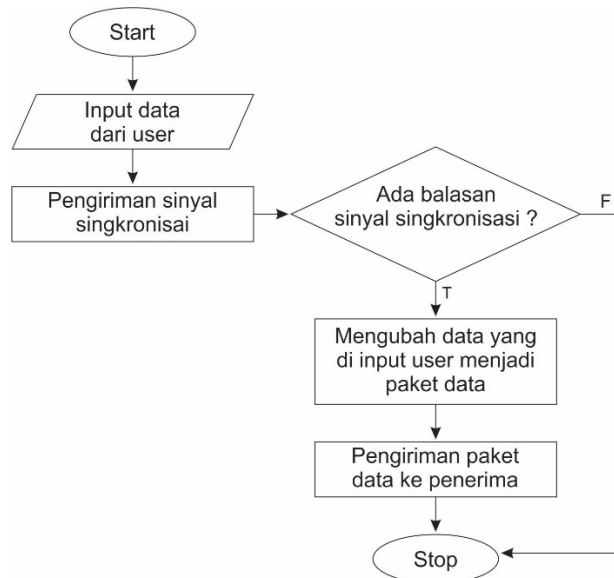
Setelah semua tahap dalam instalasi I2C LCD dilakukan berikutnya adalah mencoba komunikasi I2C dengan langsung mengirimkan data ke LCD, hal yang dapat ditampilkan dalam LCD adalah semua tipe data seperti *string*, *integer*, *double / float*, dan *char*. Perhatikan *source code* di bawah ini :

```
LCD1.print("LCD READY");  
LCD1.print(d_barang);  
LCD1.print(d_harga);
```

### 3.4 Perancangan Perangkat Lunak

Dari perancangan sistem di atas, selain perencanaan *hardware* juga dibutuhkan perancangan perangkat lunak untuk menjalankan perancangan *hardware* yang telah dibuat, perangkat lunak ini berisi tentang informasi yang akan dikirimkan ke *hardware* melalui jaringan.

### 3.4.1 Algoritma Pengiriman Data



**Gambar 3.8** Flowchart Algoritma Pengiriman Data

Seperti yang dijelaskan oleh Gambar 3.6, data yang diinputkan *user* akan dibuatkan sebuah paket data dan dikirimkan ke penerima, namun sebelum mengirimkan *software* ini akan mengirimkan sinyal sinkronisasi yang fungsinya untuk memastikan bahwa penerima itu aktif. Jika penerima tidak dapat merespon sinyal sinkronisasi maka penerima dianggap tidak aktif dan data tidak di kirim.

Adapun informasi data yang akan dimuat dalam perangkat lunak atau aplikasi antara lain :

1. IP *address* tujuan.
2. I2C tujuan.
3. Nama Barang.
4. Harga barang.

Tampilan perangkat lunak dapat dilihat pada Gambar di bawah ini :

**Gambar 3.9** Tampilan Perangkat Lunak

a. Sinkronisasi

Pada sinkronisasi penulis memanfaatkan fasilitas *ping* untuk mengetahui apakah alamat destinasi sedang aktif atau sedang dalam keadaan mati. Pada fungsi *ping software* akan mengirimkan sebuah paket ICMP ke alamat yang dituju, lalu jika alamat tujuan hidup maka *hardware* alamat tersebut akan mengirimkan sinyal *echo* atau sinyal *feedback*. Sinyal *feedback* ini yang akan menjadi acuan apakah proses dilanjutkan atau tidak.

Untuk menggunakan fasilitas *ping* awal mula penulis mendaftarkan *library network information* terlebih dahulu, ini digunakan untuk memunculkan fungsi *ping* dalam program.

*using System.Net.NetworkInformation;*

Untuk *ping* sendiri setelah *library* terdaftar penulis harus membuat variabel terlebih dahulu untuk menggantikan variabel *ping* di sini penulis menggunakan variabel *isping* sebagai pemanggil, lalu pada baris kedua penulis membuat variabel baru yang diisi perintah untuk melakukan *ping* pada “*isping.send()*” hasil dari *ping* akan ditampung oleh variabel *reply*, perhatikan *source code* berikut:

*Ping isping = new Ping();*

*PingReply reply = isping.Send(comboBox1.Text);*

Pada saat ini hasil dari *ping* masih ditampung pada *reply* dan *reply* bukan *boolean* yang artinya datanya masih belum bisa dibaca. Untuk solusi saat ini, penulis menambahkan variabel untuk menyimpan hasil dari *ping*, dimana penulis membuat variabel baru berupa *pinging* yang digunakan untuk menampung status dari hasil *reply*. Perhatikan *source code* berikut:

```
bool pinging = false;

pinging = reply.Status == IPStatus.Success;
```

Berikut adalah *source code* lengkap pada bagian sinkronisasi:

```
Ping isping = new Ping();
bool pinging = false;

try
{
    PingReply reply = isping.Send(comboBox1.Text);
    pinging = reply.Status == IPStatus.Success;
}
catch
{
    pinging = false;
}

if (pinging == false)
{
    MessageBox.Show("Check your IP address Destination...!!!", "Warning");
}
```

#### b. Pengiriman data

Pengiriman data yang dilakukan menggunakan protokol telnet, karena protokol telnet sendiri adalah protokol yang dibuat untuk keperluan komunikasi 2 arah antara server dan *client* dengan memanfaatkan *virtual terminal*. Dipilihnya telnet untuk protokol komunikasi yang digunakan adalah karena telnet support kepada wemos tidak seperti FTP yang harus melalui media parsing *web*. Pada Sendiri pengiriman data berbasis text lebih cepat karena pengiriman bersifat langsung, tidak melalui media *web* jadi pengiriman lebih cepat dengan data yang



kecil. Pengiriman data sendiri akan dikirimkan melalui media jaringan *Wi-Fi* yang menghubungkan antara Wemos dan server.

Untuk menggunakan fasilitas telnet awal mula penulis mendaftarkan *library net* dan *net.sockets* terlebih dahulu, *sockets* ini merupakan mekanisme komunikasi yang memungkinkan terjadinya pertukaran data antarprogram atau proses baik pada satu mesin atau antarmesin. Fasilitas *sockets* tersedia pada sistem operasi yang berkomunikasi dengan protokol TCP/IP. Seperti pada *source code* berikut :

```
Using System.Net;
using System.Net.Sockets;
```

Sedangkan untuk menggunakan fungsi telnet setelah mendaftarkan library berikutnya yang dilakukan penulis adalah pembuatan 2 variabel penting yaitu variabel *socket* dan *IpEndPoint*. Dimana *socket* berfungsi untuk membuat jalur dan membuka tutup akses, lalu *IpEndPoint* digunakan untuk pengalamatan jalur yang akan dibuka dan dikirimkan data oleh *socket*. Seperti pada *source code* berikut :

```
Socket sck = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
IPEndPoint localEndPoint = new IPEndPoint(IPAddress.Parse(combobox1.Text),
int.Parse("23"));
```

Setelah variabel terbentuk, penulis menyiapkan data yang akan dikirimkan melalui program ini, format yang digunakan oleh pengirim dalam mengirimkan data adalah *alamat\_i2c*, *nama\_barang*, *harga\_barang*. Setelah format data tersusun data akan di-convert atau di-encoding menjadi nilai ASCII karena pada protokol TCP/IP yang dikirimkan adalah data *byte*, ini juga dapat digunakan sebagai enkripsi data. Seperti pada *source code* berikut :

```
string nilai = combobox2.Text + "," + textBox4.Text + ",Rp " + textBox5.Text;
byte[] data = Encoding.ASCII.GetBytes(nilai);
```

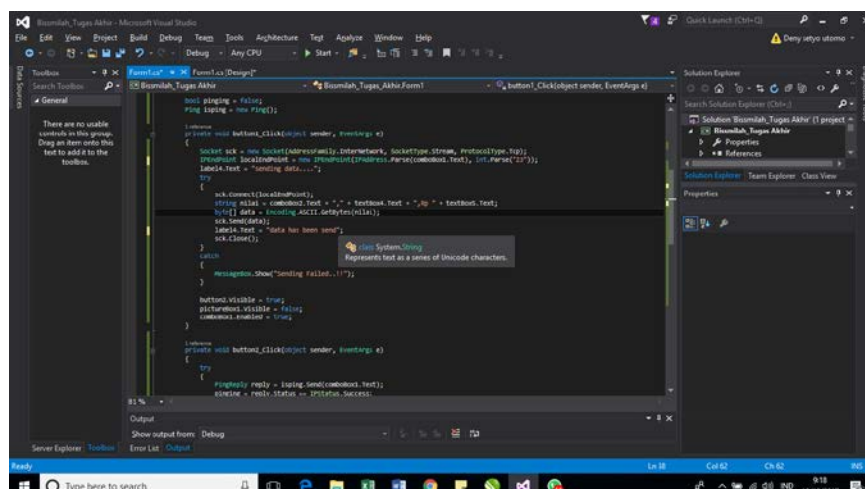
setelah data terbentuk maka penulis siap untuk mengirimkan paket data tersebut, yang berperan di pengiriman data ini adalah *socket* yang akan membuka dan mengirimkan data lalu menutup jalur komunikasi. Perhatikan *source code* berikut :

```
sck.Connect(localEndPoint);
sck.Send(data);
sck.Close();
```

Berikut adalah *source code* lengkap pada bagian pengiriman data :

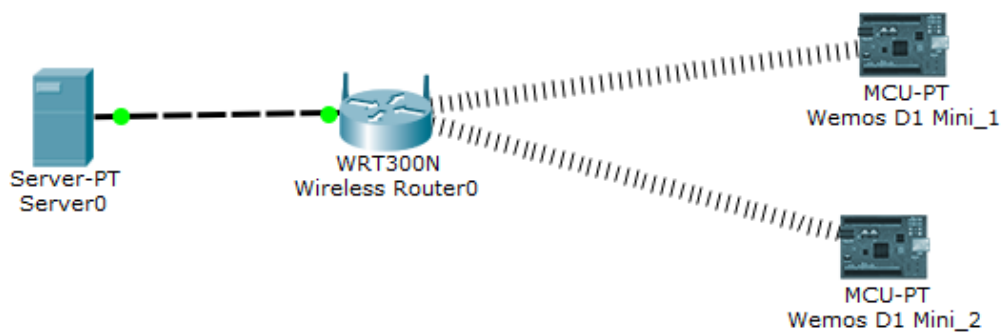
```
Socket sck = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
IPEndPoint localEndPoint = new IPEndPoint(IPAddress.Parse(combobox1.Text),
int.Parse("23"));
label4.Text = "sending data....";
try
{
    sck.Connect(localEndPoint);
    string nilai = combobox2.Text + "," + textBox4.Text + ",Rp " + textBox5.Text;
    byte[] data = Encoding.ASCII.GetBytes(nilai);
    sck.Send(data);
    label4.Text = "data has been send";
    sck.Close();
}
catch
{
    MessageBox.Show("Sending Failed!!!");
}
```

Hasil dari penulisan *source code* dapat dilihat pada gambar 3.10 *source code* ditulis pada software Visual Studio.



**Gambar 3.10** Tampilan *Source Code* Pada Visual Studio.

### 3.5 Perancangan Topologi Jaringan



**Gambar 3.11** Topologi Jaringan Sistem

Topologi jaringan yang digunakan dalam membuat sistem ini adalah seperti Gambar 3.11. Menggunakan 1 *Router Access point* utama sebagai pengatur lalu lintas data dan penyebar sinyal WLAN untuk *client* koneksi melalui WLAN.

Model pengalamatan IP yang digunakan adalah antara *router* dan komputer menggunakan alamat jaringan 192.168.100.0 dengan *subnet mask* 255.255.255.252 / 30. Pada sisi jaringan server hanya akan disediakan IP 2 buah yaitu untuk *router* dan komputer saja. Ini digunakan untuk mengamankan jaringan server agar bila ada yang ingin terhubung ke jaringan server tidak bisa. Pada sisi *client* menggunakan 1 alamat jaringan yaitu 192.168.0.0 dan *subnet mask*-nya adalah 255.255.192.0 / 18. dengan subnet tersebut *router* akan menyediakan alamat IP untuk *client* sebanyak 16.382 *hosts* mulai dari 192.168.1.2 sampai dengan 192.168.64.254. Hal ini digunakan untuk mempermudah komunikasi antarWemos.

Untuk mewujudkan pembagian IP seperti di atas, maka dibutuhkan tambahan *access point* untuk memperluas jangkauan WLAN. *Access point* ini berfungsi untuk membagi data dan sinyal dari 1 *node* ke banyak *node*. Dengan

*access point* ini 1 *node* dari *router* akan disebarkan ke *access point* yang lain. *Access point* ini digunakan untuk menghubungkan koneksi antara Wemos ke jaringan. Setiap Wemos yang akan terhubung ke jaringan harus melewati *access point* yang SSID-nya sama dengan IP yang akan ditentukan secara *static*.

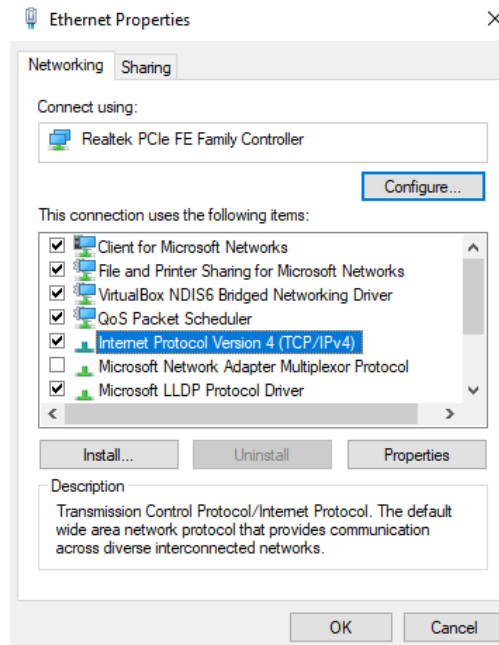
### 3.5.1 Konfigurasi IP Server

Untuk mewujudkan jaringan seperti Gambar 3.11, dibutuhkan konfigurasi pada bagian server terutama pada bagian IP Address dan *subnet mask*. Dalam implementasinya server diberikan IP 192.168.100.1 dengan *subnet mask* 255.255.255.252 / 30. Untuk melakukan pengaturan IP tersebut menggunakan windows adalah dengan cara masuk ke *control panel* komputer lalu masuk ke bagian *network and internet* pilih bagian *network connections*. Maka akan muncul Gambar seperti Gambar 3.12.



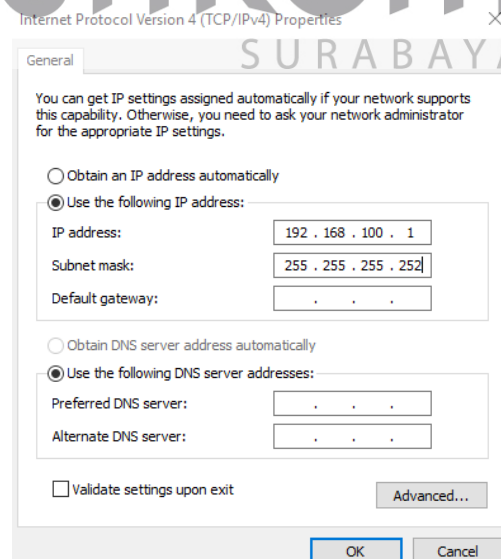
**Gambar 3.12** Tampilan *Network connections*

Setelah muncul seperti Gambar 3.12 pilih *ethernet* yang tersambung ke *router* langsung. Klik kanan pada *ethernet* yang terpilih lalu klik *properties*. Akan muncul seperti di Gambar 3.13.



**Gambar 3.13** Tampilan *Ethernet Properties*

Setelah muncul seperti di Gambar 3.13 klik 2 kali pada *Internet Protocol Version 4 (IPv4)* atau klik 1 kali lalu klik *button properties* akan muncul seperti Gambar 3.14.



**Gambar 3.14** Tampilan *Ipv4 Properties*

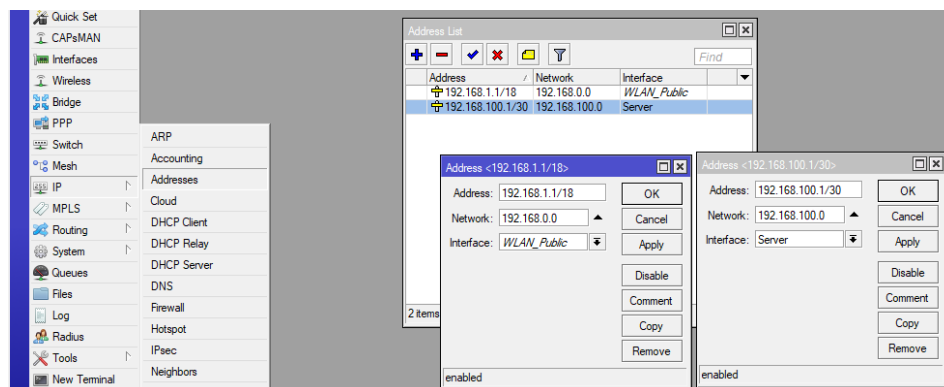
Pada Gambar 3.14 akan dilakukan *setting IP Address* pada topologi ini PC digunakan sebagai server maka yang akan diisi adalah pada bagian *IP Address* dan *Subnet mask*. IP diisi dengan 192.168.100.2 dan *subnet mask* adalah 255.255.255.252 / 30.

Setelah langkah tersebut selesai maka IP komputer kita adalah 192.168.100.2 selanjutnya klik *ok* lalu tutup *windows*.

### 3.5.2 Configurasi Routing dan IP Router

Untuk menghubungkan antara jaringan Wemos dan jaringan server diperlukan sebuah *router* agar kedua jaringan tersebut dapat di-*routing*. Pada penelitian ini *router* yang digunakan oleh penulis adalah *router* Mikrotik. Untuk mewujudkan topologi jaringan seperti pada Gambar 3.11 *router mikrotik* harus di-*setting* agar dapat membuat jaringan dan menghubungkan jaringan.

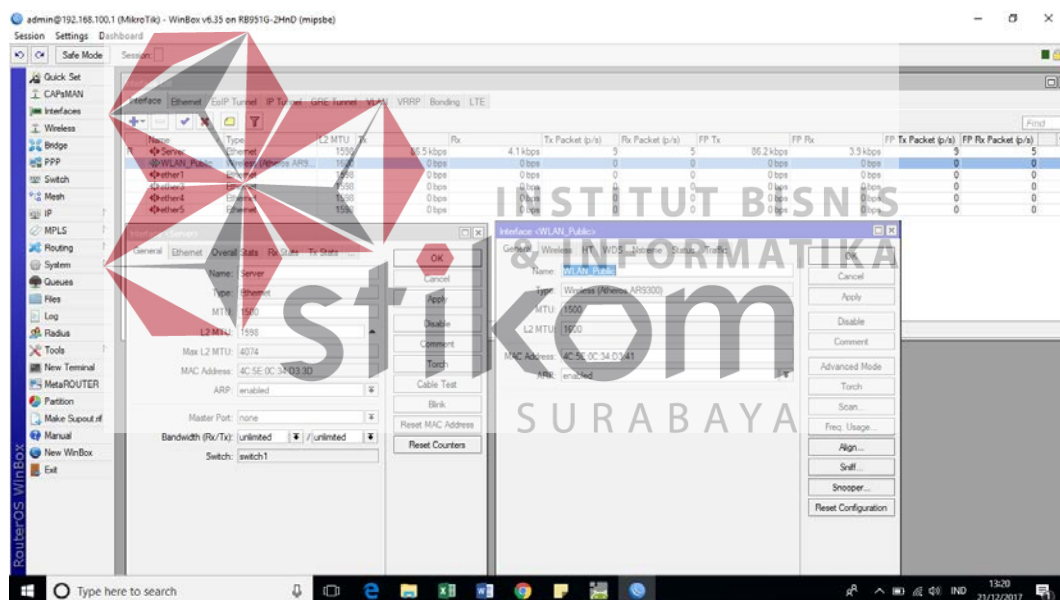
Adapun langkah awal yang harus dilakukan adalah melakukan *setting IP mikrotik* pada Gambar 3.11 jaringan dibagi menjadi 2 yaitu jaringan LAN untuk server yang memiliki alamat jaringan 192.168.100.0 dan jaringan WLAN untuk Wemos yang memiliki alamat jaringan 192.168.0.0.



**Gambar 3.15** Setting IP LAN dan WLAN Mikrotik

Pada Gambar 3.15 dapat dilihat bahwa LAN di-setting dengan IP 192.168.100.1/30 dan WLAN di-setting IP 192.168.1.1/18. Artinya LAN hanya akan menyediakan 2 *host* 1 digunakan untuk *gateway* dan satunya lagi digunakan untuk komputer. Sedangkan pada WLAN menyediakan 16.382 *host*.

Setelah IP di-setting selanjutnya adalah melakukan konfigurasi *interface* dimana konfigurasi ini berguna untuk membedakan nama *interface* agar lebih mudah dipahami. Juga untuk menyalakan *Wireless* Mikrotik karena setingan *default* adalah *Wireless* dalam keadaan mati.

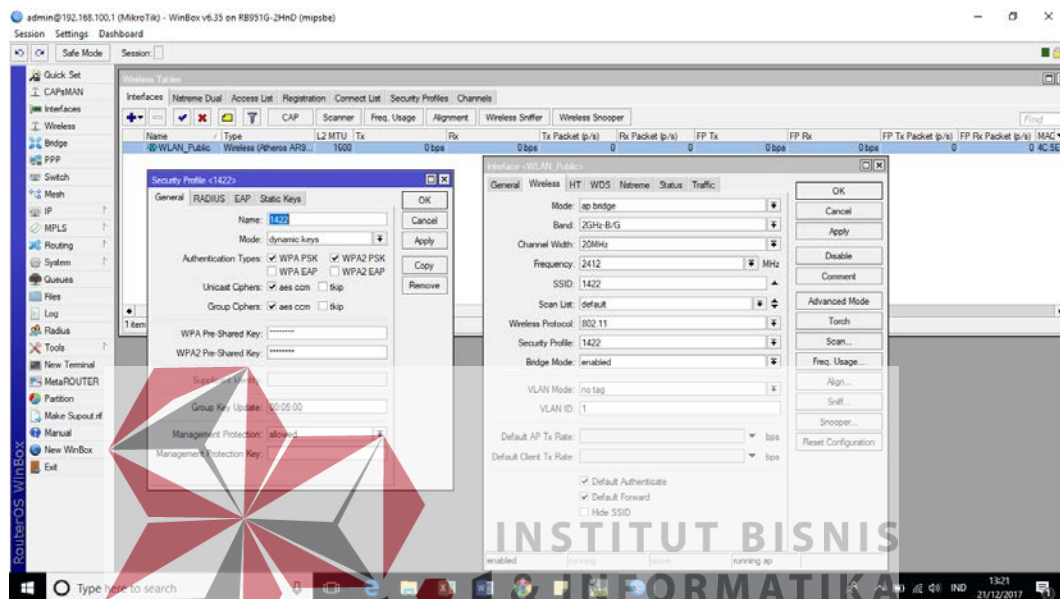


**Gambar 3.16** Konfigurasi *interface* router Mikrotik

Pada Gambar 3.16 dapat dilihat bahwa *interface* LAN diberi nama server dan pada jaringan WLAN diberi nama WLAN\_Public. Lalu untuk mengaktifkan *Wireless* Mikrotik cukup mengklik *interface* WLAN\_Public lalu klik *icon* centang maka *Wireless* mikrotik sudah aktif.



Setelah WLAN aktif, selanjutnya adalah melakukan *setting* konfigurasi WLAN agar dapat digunakan sebagai *access point*, juga melakukan konfigurasi keamanan WLAN agar tidak mudah dimasuki orang dengan cara memberi keamanan pada jaringan WLAN.



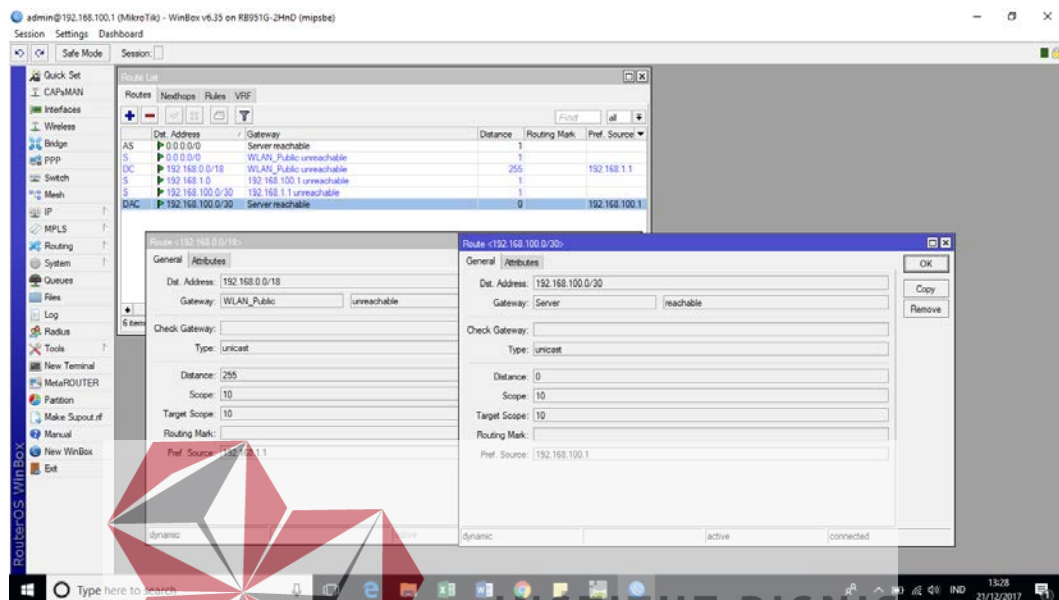
Gambar 3.17 Konfigurasi WLAN dan Keamanan WLAN

Pada Gambar 3.17 dapat dilihat bahwa pengaturan WLAN dibuat *mode ap bridge* agar dapat menjadi *access point* dan nama SSID yang dibuat adalah 1422, artinya nantinya *client* akan mencari *access point* dengan nama 1422 untuk masuk ke jaringan WLAN.

Dari Gambar 3.17 pula dapat dilihat bahwa pengamanan *WiFi* menggunakan WPA PSK dan WPA2 PSK dan diberi *password*, artinya *client* yang ingin memasuki *access point* ini diharuskan mengisi *password* sebelum masuk. Ini berguna untuk pengamanan agar yang tidak mempunyai kepentingan tidak dapat masuk ke dalam jaringan.



Setelah WLAN dan LAN aktif, ini tidak dapat langsung berkomunikasi karena berada di jaringan berbeda. Adapun hal yang harus dilakukan selanjutnya adalah melakukan *routing* antara jaringan WLAN dan Jaringan LAN.



Gambar 3.18 Setting routing Mikrotik

Pada Gambar 3.18 dapat dilihat pengaturan *routing* yaitu dengan mengisi *destinasi address* adalah alamat jaringan yang dituju dan *gateway* adalah alamat yang akan kita lewati menuju ke jaringan yang kita tuju. Untuk *routing* server adalah *destinasi address* menuju WLAN\_Publik dan untuk WLAN\_Publik adalah *destinasi address* menuju server. Kedua jaringan ini harus saling di-*routing* agar dapat saling berkomunikasi.

### 3.5.3 Konfigurasi IP Static pada Wemos D1 mini

Pengalamatan IP pada Wemos murni menggunakan *source code* untuk konfigurasinya yang harus ditentukan pertama adalah alamat IP *Address local* Wemos, IP *subnet mask*, dan IP *gateway*. Pada penelitian ini penulis

menggunakan alamat jaringan adalah 192.168.0.0 dan IP *gateway* 192.168.1.1 lalu menyediakan *host* sebanyak 16.382 artinya bahwa IP yang disediakan untuk Wemos adalah sebanyak 16.379. Hasil tersebut adalah setelah *host* dikurangi 3 karena untuk alamat jaringan, alamat *broadcast*, dan alamat *gateway*. Untuk *range* alamat IP yang dapat digunakan adalah antara 192.168.1.2 – 192.168.64.254. Perhatikan *source code* berikut :

```
IPAddress localIP(192.168.1.100);  
IPAddress subnet(255, 255, 192, 0);  
IPAddress gateway(192.168.1, 1);
```

Setelah menentukan IP Wemos yang selanjutnya dilakukan adalah mengubah konfigurasi yang ada di Wemos. Karena dengan hanya mengisi IP tidak dapat mengubah IP Wemos, harus ada yang mengeksekusi perubahan

```
WiFi.config(localIP, gateway, subnet);
```

Setelah melakukan perintah *WiFi.config()* aka IP Wemos sudah berganti, bila prosedur di atas tidak dilakukan maka Wemos memiliki IP DHCP.

## BAB IV

### HASIL PENGUJIAN DAN PENGAMATAN

Dalam bab ini penulis akan menguraikan dan menjelaskan beberapa hasil pengujian dan hasil penelitian tugas akhir ini. Pengujian yang dilakukan meliputi pengujian perangkat lunak dan kinerja keseluruhan sistem, serta analisa pengiriman data dari PC ke Wemos lalu dari Wemos ke LCD. Perlengkapan alat yang digunakan dalam pengujian ini dapat dilihat di Gambar 4.1.



**Gambar 4.1** Tampilan peralatan yang digunakan untuk pengujian

#### 4.1 Pengujian Wemos

Dalam pengujian Wemos dilakukan dengan menuliskan program pada *sketch* Arduino IDE lalu di-*upload* pada Wemos.

#### 4.1.1 Tujuan

Pengujian ini dilakukan untuk memastikan Wemos masih berfungsi dengan baik dan tidak dalam keadaan rusak.

#### 4.1.2 Alat yang digunakan

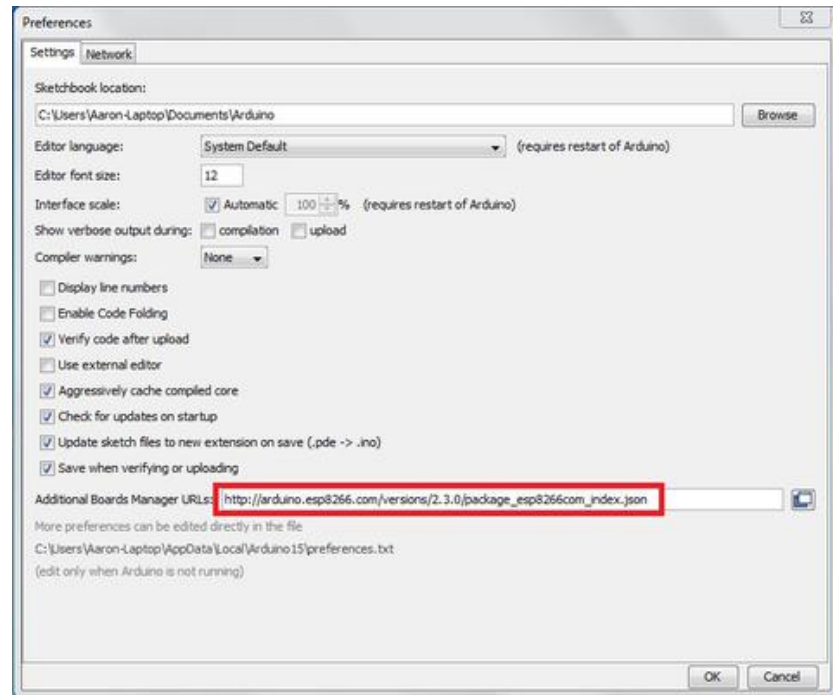
Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

- a. *Power supply 5V*
- b. Kabel data USB *mikro*
- c. Wemos D1 Mini
- d. Komputer / Laptop
- e. *Software* Arduino IDE

#### 4.1.3 Prosedur pengujian

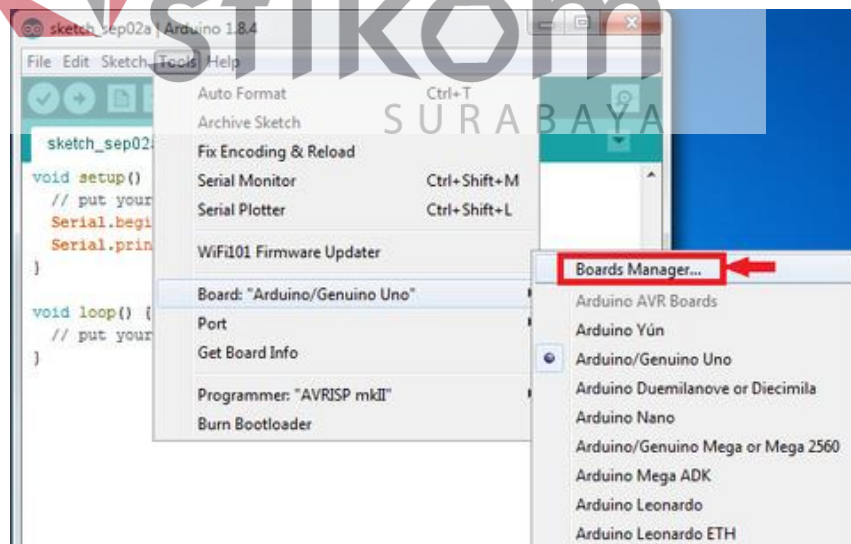
Prosedur pengujian :

1. Nyalakan komputer / laptop yang akan digunakan untuk menguji Wemos.
2. Buka *software* Arduino IDE.
3. Lakukan instalasi *library* Wemos pada Arduino IDE.
  - a. Untuk mengatur *download boards* Wemos D1 Mini masuk pada menu *file* lalu klik *preferences*, masukkan alamat *url* berikut   
[http://arduino.esp8266.com/versions/2.3.0/package\\_esp8266com\\_index.js](http://arduino.esp8266.com/versions/2.3.0/package_esp8266com_index.js)   
[on](#) kedalam kolom *Additional Boards Manager Urls* lalu klik OK, perhatikan Gambar 4.1.



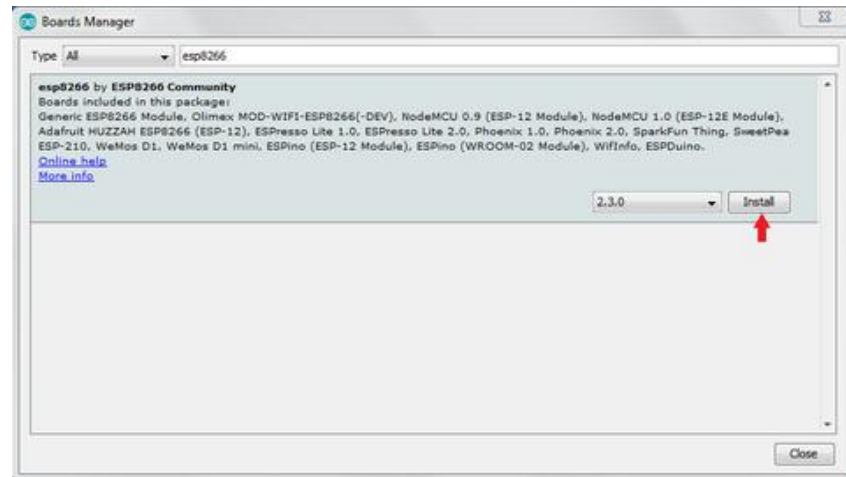
Gambar 4.2 File preferences

- b. Setelah selesai masukan URL kemudia buka *tools* pilih *board* lalu klik *board manager*. Seperti Gambar 4.2.



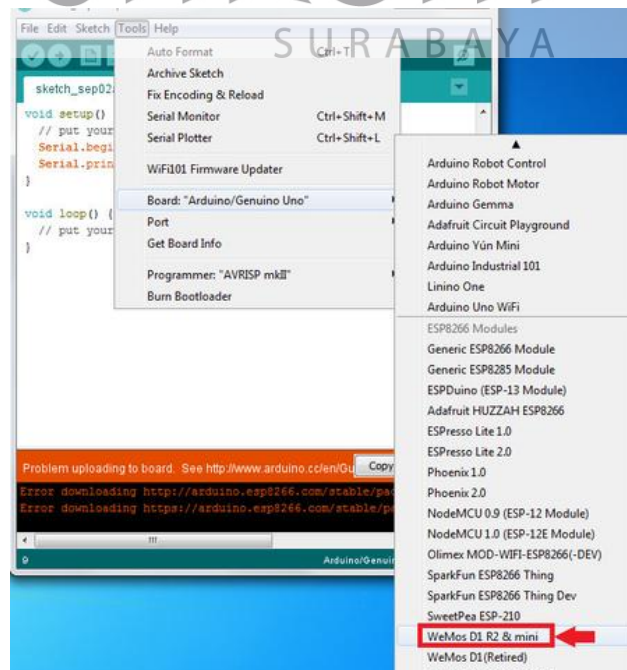
Gambar 4.3 Boards Manager

- c. Setelah muncul window *board manager*, tuliskan pada kolom “esp8266” lalu tekan *enter*, setelah pencarian selesai klik tombol *install*, lalu tunggu hingga proses *download* dan *install* selesai. Perhatikan Gambar 4.3.



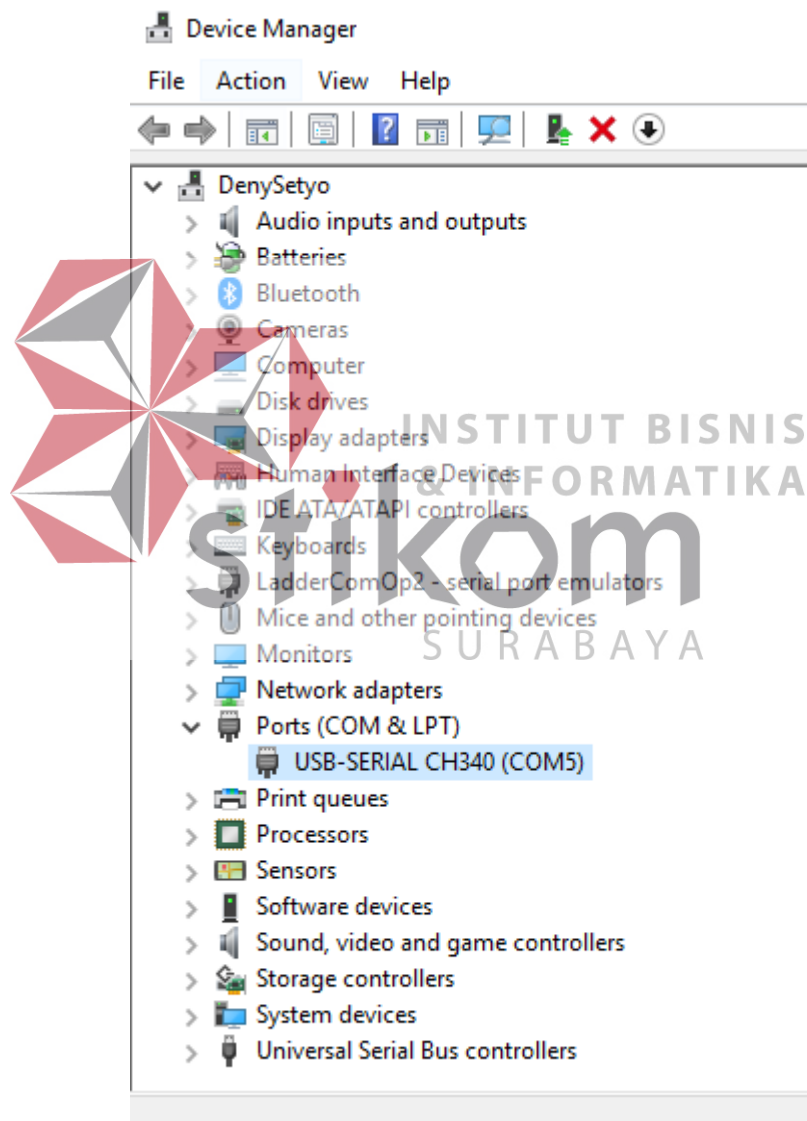
**Gambar 4.4** Pencari dan *install boards* Wemos.

- d. Setelah instalasi selesai silahkan cek pada menu *tools* pilih *board* lalu cari Wemos D1 Mini, bila Arduino IDE sudah support Wemos dan memiliki *library board*-nya. Perhatikan Gambar 4.4.



**Gambar 4.5** Pengecekan *boards* Wemos D1 Mini Sudah terinstall

4. Masukkan kabel data USB *mikro* pada Wemos D1 Mini dan sisi USB dihubungkan kepada USB laptop.
5. Salin *Source code* dengan judul “pengujian Wemos menggunakan *led blink*” pada lampiran
6. Buka *device manager* pada laptop, periksa COM *port* Wemos berada di COM nomor berapa. Seperti pada Gambar 4.5

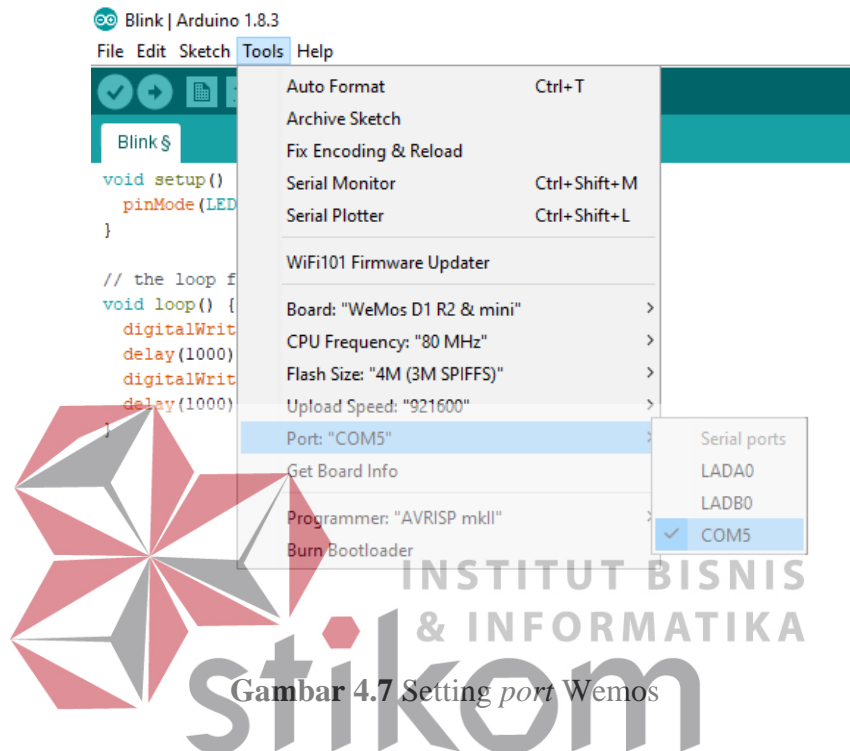


**Gambar 4.6** Tampilan *device manager*

Dari Gambar 4.5 dapat di lihat bahwa Wemos berada di COM 5 pada laptop penulis.

7. Sesuaikan *port* Wemos pada Arduino IDE dengan *device manager*.

Perhatikan Gambar 4.6.

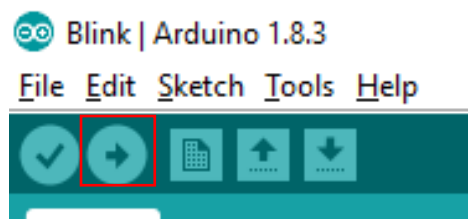


Gambar 4.7 Setting *port* Wemos

Dari Gambar 4.6 dapat dilihat bahwa pada Arduino IDE sudah di-*setting* *port* komunikasi Wemos berada di COM5.

8. *Upload* program yang telah dibuat menuju ke Wemos dengan meng-klik tombol *upload* yang bergambar panah ke kanan di bawah *menu*. Perhatikan

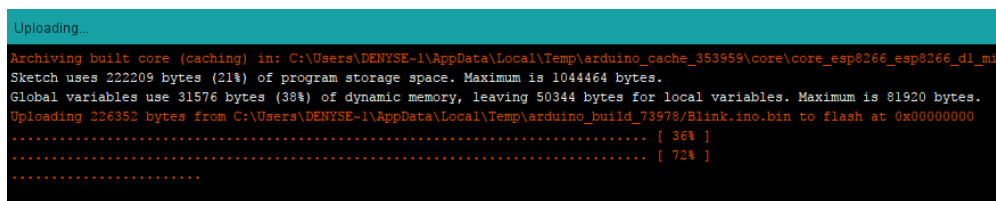
Gambar 4.7



Gambar 4.8 Icon *Upload*



Pada Gambar 4.7 dapat dilihat bahwa setelah *icon upload* di-klik maka Arduino IDE melakukan proses *upload* program menuju ke Wemos melewati *port* komunikasi.



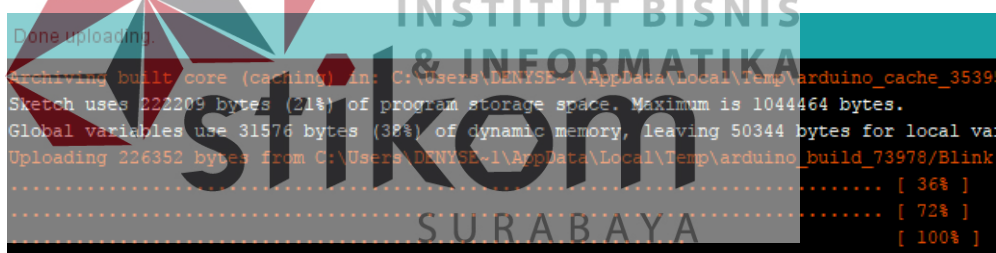
```

Uploading...
Archiving built core (caching) in: C:\Users\DENYSE-1\AppData\Local\Temp\arduino_cache_353959\core\core_esp8266_esp8266_d1_m1
Sketch uses 222209 bytes (21%) of program storage space. Maximum is 1044464 bytes.
Global variables use 31576 bytes (38%) of dynamic memory, leaving 50344 bytes for local variables. Maximum is 81920 bytes.
Uploading 226352 bytes from C:\Users\DENYSE-1\AppData\Local\Temp\arduino_build_73978/Blink.ino.bin to flash at 0x00000000
..... [ 36% ]
..... [ 72% ]
.....
  
```

**Gambar 4.9** Proses *upload* program ke Wemos

#### 4.1.4 Hasil Pengujian

Setelah melakukan pengujian kepada Wemos didapatkan hasil seperti pada Gambar 4.9 dan Gambar 4.10.

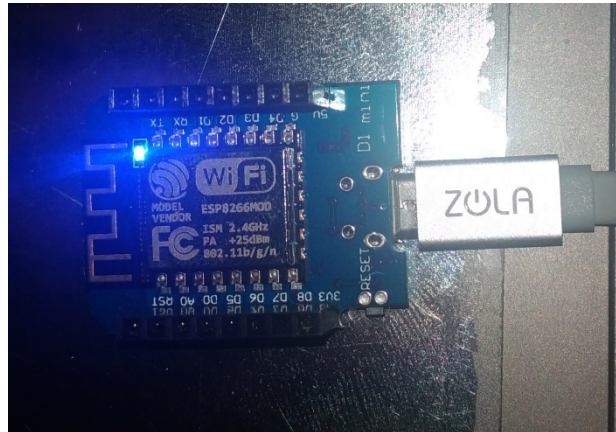


```

Done uploading.
Archiving built core (caching) in: C:\Users\DENYSE-1\AppData\Local\Temp\arduino_cache_3539
Sketch uses 222209 bytes (21%) of program storage space. Maximum is 1044464 bytes.
Global variables use 31576 bytes (38%) of dynamic memory, leaving 50344 bytes for local va
Uploading 226352 bytes from C:\Users\DENYSE-1\AppData\Local\Temp\arduino_build_73978/Blink
..... [ 36% ]
..... [ 72% ]
..... [ 100% ]
  
```

**Gambar 4.10** *Upload* program ke Wemos sukses

Pada Gambar 4.9 menunjukkan bahwa program yang di-*upload* menuju Wemos berhasil sampai 100%, hal ini menunjukkan Wemos dalam keadaan sudah dapat dimasuki program.



**Gambar 4.11** Hasil dari *upload* program

Pada Gambar 4.10 menunjukkan bahwa program yang telah dibuat dapat berjalan sesuai yang diinginkan, ditunjukkan dengan menyala lampu led yang berada di Wemos. Ini menunjukkan bahwa Wemos dapat berjalan sesuai dengan mestinya, juga dapat disimpulkan bahwa Wemos dalam keadaan baik dan siap digunakan.

## 4.2 Pengujian *Scanning Access point* oleh Wemos

Dalam pengujian *scanning access point* oleh Wemos dilakukan dengan menuliskan program pada *sketch* Arduino IDE yang memerintahkan Wemos untuk mencari *access point* yang berada di radius jangkauan *WiFi* Wemos.

### 4.2.1 Tujuan

Pengujian ini dilakukan untuk memastikan Wemos dapat mencari *access point* yang dituju dan memastikan keadaan *WiFi* Wemos dalam keadaan baik.

#### 4.2.2 Alat yang digunakan

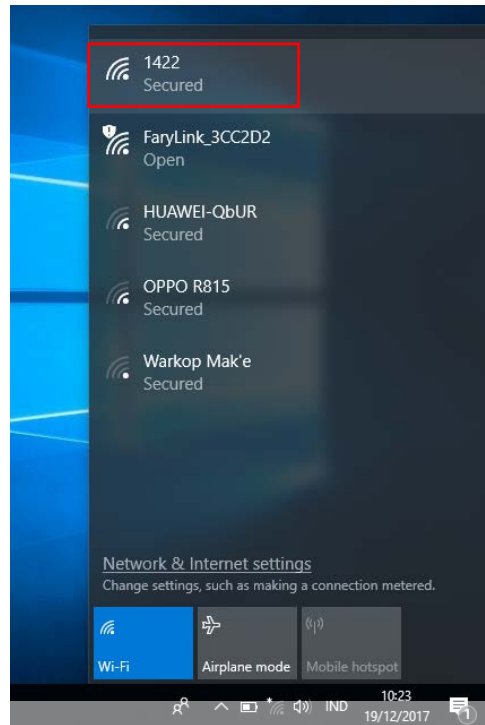
Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

- a. *Power supply 5V*
- b. *Akses Point*
- c. *Software Arduino IDE*
- d. Laptop / Komputer
- e. Kabel data *mikro* USB
- f. Wemos D1 Mini

#### 4.2.3 Prosedur pengujian

Prosedur pengujian :

1. Nyalakan *access point*
2. Aktifkan laptop dan nyalakan *WiFi* pada laptop.
3. Buka *WiFi* dan pastikan jaringan *Wireless* yang disebar oleh *access point* terlihat oleh laptop sebelum melakukan pengujian seperti pada Gamabr 4.11.



**Gambar 4.12** Tersedia jaringan WLAN

4. Tancapkan Kabel data USB mikro pada Wemos D1 Mini dan sisi USB dihubungkan kepada USB laptop.
5. Buka *software* Arduino IDE.
6. Salin *Source code* dengan judul “Pengujian *scanning* Access point Wemos” pada lampiran
7. Pastikan *port* COM untuk komunikasi Wemos antara di Arduino IDE dan *device manager* sama.
8. *Upload* program yang telah dibuat menuju ke Wemos.
9. Setelah *upload* selesai buka *serial monitor* yang berada di sebelah kanan pada Arduino IDE yang memiliki *icon* seperti kaca pembesar, lihat pada Gambar 4.12



**Gambar 4.13** Icon *serial monitor*

#### 4.2.4 Hasil Pengujian

Setelah melakukan pengujian *scanning Access point* Wemos maka didapatkan hasil pada *serial monitor* seperti pada Gambar 4.13.



**Gambar 4.14** Hasil *Scanning access point* oleh Wemos

Pada Gambar 4.13 menunjukkan hasil dari pantauan *serial monitor* Wemos melalui *port* COM 5, dari gambar tersebut membuktikan Wemos dapat menemukan *access point* yang telah disebar oleh *access point* dan dari hasil ini juga dapat disimpulkan bahwa kondisi *WiFi* yang dimiliki Wemos bekerja dengan baik.

#### 4.3 Pengujian menghubungkan *WiFi* Wemos pada *access point*

Dalam pengujian menghubungkan *WiFi* Wemos pada *access point* ini dilakukan dengan menuliskan program pada *sketch* Arduino IDE yang

memerintahkan Wemos untuk menghubungkan diri kepada *access point* yang telah ditentukan.

#### 4.3.1 Tujuan

Pengujian ini dilakukan untuk memastikan Wemos dapat menghubungkan diri kepada *access point* yang ditentukan serta dapat berkomunikasi di jaringan yang dibuat oleh *access point* tersebut.

#### 4.3.2 Alat yang digunakan

Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

- a. *Power supply 5V*
- b. *Akses Point*
- c. *Software Arduino IDE*
- d. Laptop / Komputer
- e. Kabel data *mikro* USB
- f. Wemos D1 Mini



#### 4.3.3 Prosedur pengujian

Prosedur pengujian :

1. Nyalakan *access point*
2. Aktifkan laptop dan nyalakan *WiFi* pada laptop.
3. Hubungkan laptop pada *access point* yang sudah disediakan.
4. Buka *command prompt*.
5. Tuliskan “ipconfig” lalu tekan *enter* pada *command prompt*.

6. Lihat IP yang didapatkan oleh laptop dengan memperhatikan adapter *WiFi* terdapat Ipv4 Address yang menjadi alamat IP komputer yang diberikan oleh *access point*. Perhatikan Gambar 4.14.

```

C:\Users\DenySetyo>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 3:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 13:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::7502:2a55:611:e643%12
    IPv4 Address. . . . . : 192.168.1.252
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

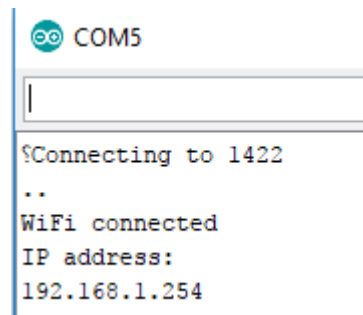
Ethernet adapter Bluetooth Network Connection:
  
```

Gambar 4.15 IP laptop yang didapat dari *access point*

Dari Gambar 4.14 dapat dilihat bahwa setelah menuliskan “ipconfig” lalu menekan *enter* akan muncul *IP configuration*. Lalu dari gambar tersebut juga dapat dilihat bahwa laptop mendapatkan IP 192.168.1.252.

7. Tancapkan Kabel data USB *mikro* pada Wemos D1 Mini dan sisi USB dihubungkan kepada USB laptop.
8. Buka *software* Arduino IDE.
9. Salin *Source code* dengan judul “Pengujian *connecting WiFi to access point*” pada lampiran
10. Pastikan *port* COM untuk komunikasi Wemos antara di Arduino IDE dan *device manager* sama.

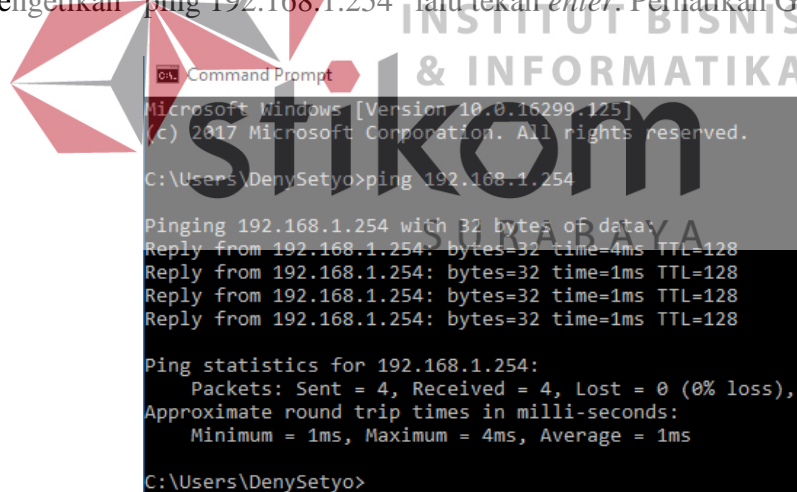
11. *Upload* program yang telah di buat menuju ke Wemos.
12. Setelah *upload* selesai buka *serial monitor*. Perhatikan Gambar 4.15.



**Gambar 4.16** Wemos terhubung ke *access point* dan menpatakan IP *address*

Dilihat dari Gambar 4.15 menunjukkan Wemos terhubung pada *access point* dan mendapatkan IP *address* 192,168.1.254.

13. Buka *command prompt*, lakukan *ping* pada IP *address* Wemos dengan mengetikan “ping 192.168.1.254” lalu tekan *enter*. Perhatikan Gambar 4.16.



**Gambar 4.17** Ping dari laptop menuju Wemos

#### 4.3.4 Hasil Pengujian

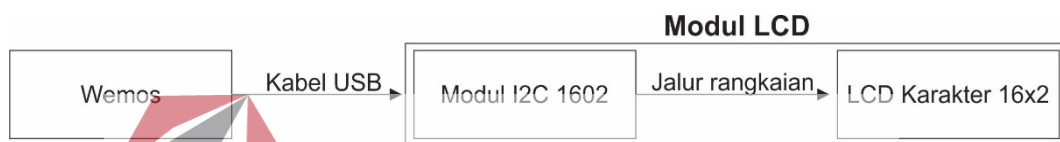
Dilihat dari Gambar 4.15 dapat disimpulkan bahwa Wemos dapat terhubung ke jaringan WLAN yang dibuat oleh *access point* dan mendapatkan IP



*address* dari *access point*, ini menunjukkan *WiFi* Wemos dalam keadaan baik dan dapat digunakan untuk komunikasi

Bila dilihat dari Gambar 4.16 dapat disimpulkan juga bahwa Wemos dapat berkomunikasi dengan laptop melalui jaringan WLAN, ini dikarenakan Wemos memberi balasan kepada laptop saat laptop mengirimkan sinyal *ping*, hal ini menunjukkan bahwa Wemos dapat berkomunikasi melalui *Wireless*.

#### 4.4 Pengujian *Scanning* dan Pengiriman I2C Wemos Menuju LCD



**Gambar 4.18** Diagram blok pengujian *scanning* dan pengiriman data I2C Wemos menuju LCD

Pada Gambar 4.17 dapat diketahui bahwa Wemos akan mengirimkan data menuju ke Modul I2C 1602 lalu oleh Modul I2C 1602 akan menerjemahkan data I2C menjadi tampilan di LCD karakter 16x2.

Dalam pengujian I2C ini dilakukan dengan menuliskan program pada *sketch* Arduino IDE yang memerintahkan Wemos mengirimkan data ke LCD menggunakan protokol I2C. Data yang dikirimkan oleh Wemos akan ditampilkan di LCD karakter 16x2.

#### 4.4.1 Tujuan

Pengujian ini dilakukan untuk memastikan Wemos D1 Mini dapat mengirimkan sinyal I2C dan dapat diterima oleh modul I2C 1602 untuk diterjemahkan menjadi tampilan *display* di LCD karakter 16x2.

#### 4.4.2 Alat yang digunakan

Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

- a. *Power supply* 5V
- b. *Software* Arduino IDE
- c. Laptop / Komputer
- d. Kabel data *mikro* USB
- e. Wemos D1 Mini
- f. Kabel USB male to male
- g. Modul I2C 1602
- h. LCD karakter 16x2



INSTITUT BISNIS  
& INFORMATIKA  
**stikom**  
SURABAYA

#### 4.4.3 Prosedur pengujian

Prosedur pengujian :

1. Nyalakan laptop atau komputer.
2. Hubungkan Kabel data USB *mikro* pada Wemos D1 Mini dan sisi USB dihubungkan kepada USB laptop.
3. Hubungkan kabel USB male to male yang satu menuju ke Wemos yang sisi lain menuju ke modul LCD

4. Buka *software* Arduino IDE.
5. Salin *Source code* dengan judul “Pengujian *scanning* I2C” pada lampiran
6. Pastikan *port* COM untuk komunikasi Wemos antara di Arduino IDE dan *device manager* sama.
7. *Upload* program yang telah dibuat menuju ke Wemos.
8. Setelah *upload* selesai buka *serial monitor*. Perhatikan Gambar 4.17.



**Gambar 4.19** Hasil *serial monitor scanning* I2C

Dari Gambar 4.17 dapat dilihat bahwa setelah Wemos melakukan *scanning* Wemos menemukan alamat I2C LCD yang terhubung dengan dirinya adalah 0x3E. Dari gambar tersebut juga bisa disimpulkan jika Wemos dapat digunakan untuk komunikasi I2C karena dapat memberi sinyal trigger dan menerima sinyal respon dari perangkat yang terhubung oleh dirinya menggunakan I2C.

9. Salin *Source code* dengan judul “Pengujian pengiriman data I2C menuju LCD” pada lampiran.

10. Samakan alamat I2C yang ada pada *source code* dengan alamat I2C yang sudah ditemukan melalui *scanning* I2C. Pada *source code* alamat I2C berada pada baris yang bertuliskan “*LiquidCrystal\_I2C lcd(0x3E,16,2);*” 0x3E adalah alamat yang harus diganti, angka 16 adalah jumlah kolom dari LCD yang digunakan, dan angka 2 adalah mewakili jumlah baris yang digunakan.
11. *Upload* program yang telah dibuat menuju ke Wemos.

#### 4.4.4 Hasil Pengujian

Dari Gambar 4.17 dapat disimpulkan bahwa Wemos dapat mengirimkan dan menerima sinyal I2C dengan contoh mencari alamat I2C perangkat yang terhubung pada dirinya.



**Gambar 4.20** Hasil pengiriman data I2C Wemos menuju LCD karakter 16x2

Dari Gambar 4.18 dapat disimpulkan pengujian pengiriman data I2C Wemos menuju LCD berhasil hal ini dikarenakan data yang dikirimkan Wemos menuju LCD ditampilkan semua secara utuh dan tidak ada data yang hilang.

#### 4.5 Pengujian Jarak Jangkauan *WiFi* Wemos



**Gambar 4.21** Diagram blok pengujian jangkauan *WiFi* Wemos

Dalam pengujian ini dilakukan percobaan pengukuran jarak jangkauan Wemos. Jika dilihat dari Gambar 4.19 digunakan 2 Wemos, 1 digunakan sebagai *access point* dan yang lain sebagai *client* akan mencari Wemos AP. Pada sisi Wemos client akan diberikan program untuk *scanning WiFi* yang dibuatkan oleh Wemos *access point*. Jika Wemos *client* masih dapat mendeteksi Wemos AP maka Wemos masih dapat terhubung. Jarak yang akan digunakan untuk pengujian tanpa penghalang adalah 3m, 5m, 10m, dan 15m. Untuk pengukuran dengan 1 penghalang menggunakan jarak 2m, 4m, 6m, 8m, dan 10m. Bila Wemos

##### 4.5.1 Tujuan

Pengujian ini dilakukan untuk mengetahui jarak jangkauan Wemos untuk menentukan jarak efektif penggunaan Wemos untuk berkomunikasi.

##### 4.5.2 Alat yang digunakan

Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

- a. 2 buah *Power supply* 5V
- b. *Software* Arduino IDE
- c. Laptop / Komputer
- d. 2 buah Kabel data *mikro* USB
- e. 2 buah Wemos D1 Mini

### 4.5.3 Prosedur pengujian

1. Nyalakan laptop atau komputer
2. Hubungkan Kabel data USB *mikro* pada Wemos D1 Mini yang akan menjadi *access point* dan sisi USB dihubungkan kepada USB laptop.
3. Buka *software* Arduino IDE.
4. Salin *Source code* dengan judul “Pengujian Wemos sebagai *access point*” pada lampiran
5. Pastikan *port* COM untuk komunikasi Wemos antara di Arduino IDE dan *device manager* sama.
6. *Upload* program yang telah dibuat menuju ke Wemos.
7. Setelah *upload* selesai buka *serial monitor*. Perhatikan Gambar 4.19.

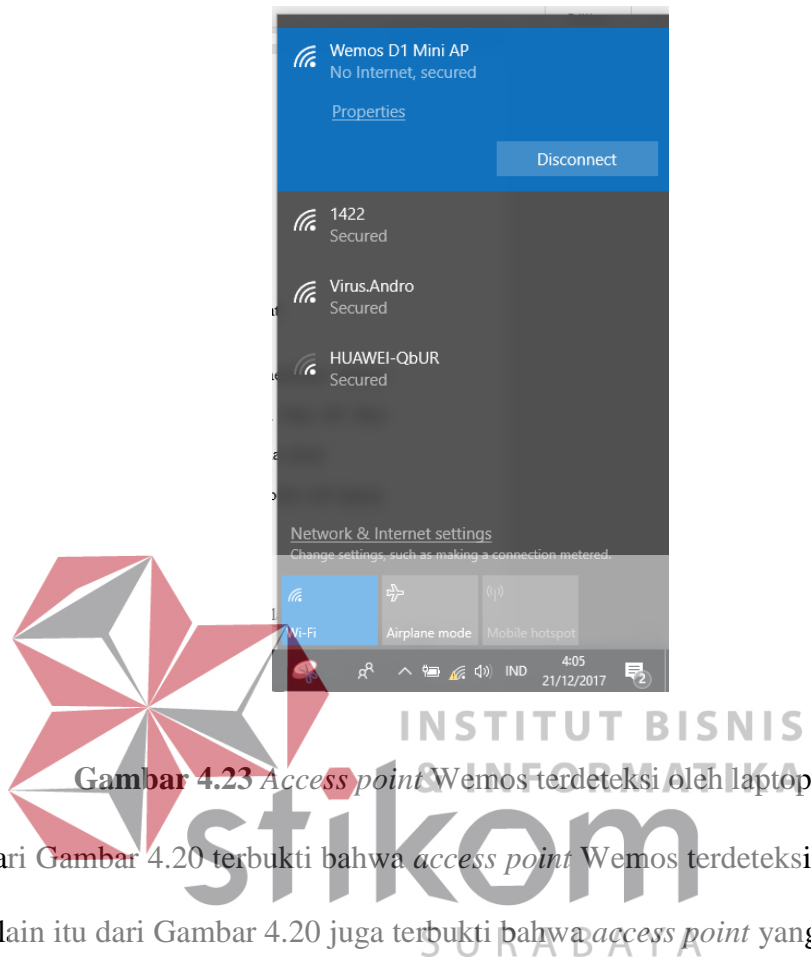


**Gambar 4.22** Wemos berhasil menjadi *access point*

Dari Gambar 4.19 menunjukkan Wemos telah berhasil membuat dirinya menjadi *access point* dengan nama SSID adalah Wemos D1 Mini AP. Dari gambar di atas juga menunjukkan bahwa Wemos siap menerima *client*.

8. Lepaskan sambungan USB Wemos yang menuju ke laptop dan pindahkan pada satu daya USB 5V sebagai pengganti daya yang awalnya berasal dari laptop.

9. Buka *WiFi* pada laptop pastikan AP Wemos dapat terdeteksi oleh *WiFi* laptop. Perhatikan Gambar 4.20.

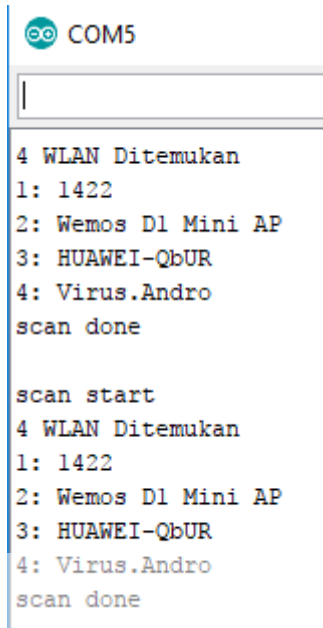


Gambar 4.23 Access point Wemos terdeteksi oleh laptop

Dari Gambar 4.20 terbukti bahwa *access point* Wemos terdeteksi oleh laptop, selain itu dari Gambar 4.20 juga terbukti bahwa *access point* yang dibuat oleh Wemos dapat dihubungkan dengan *client* pada contoh ini *client* adalah laptop.

10. Hubungkan Kabel data USB *mikro* pada Wemos D1 Mini yang akan menjadi *client* dan sisi USB dihubungkan kepada USB laptop.
11. Buka *software* Arduino IDE.
12. Salin *Source code* dengan judul “Pengujian *scanning Access point* Wemos” pada lampiran
13. Pastikan *port* COM untuk komunikasi Wemos antara di Arduino IDE dan *device manager* sama.

14. *Upload* program yang telah dibuat menuju ke Wemos.
15. Setelah *upload* selesai buka *serial monitor*. Perhatikan Gambar 4.21.



**Gambar 4.24** AP yang dibuat Wemos terlihat oleh Wemos *client*

Dari Gambar 4.21 dapat dilihat bahwa Wemos *client* dapat menemukan *access point* yang dibuat oleh Wemos AP. Dari gambar tersebut juga dapat disimpulkan bahwa antara Wemos AP dan Wemos *client* berada pada jarak yang baik dan dapat terhubung.

16. Buat jarak antara Wemos AP dan Wemos *client* sampai Wemos *client* tidak dapat menemukan Wemos AP.

#### 4.5.4 Hasil Pengujian

Dari pengujian jarak yang dilakukan oleh penulis maka penulis mencatat hasil dari percobaan di atas seperti pada Tabel 4.1.

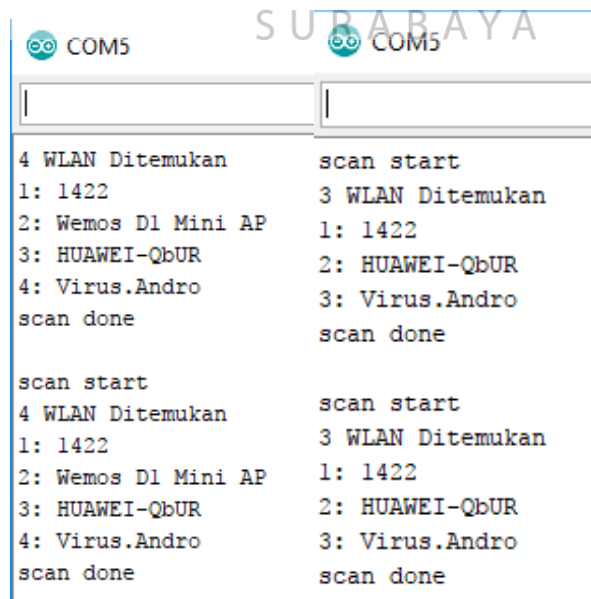


**Tabel 4.1** Hasil pengujian jarak jangkauan Wemos

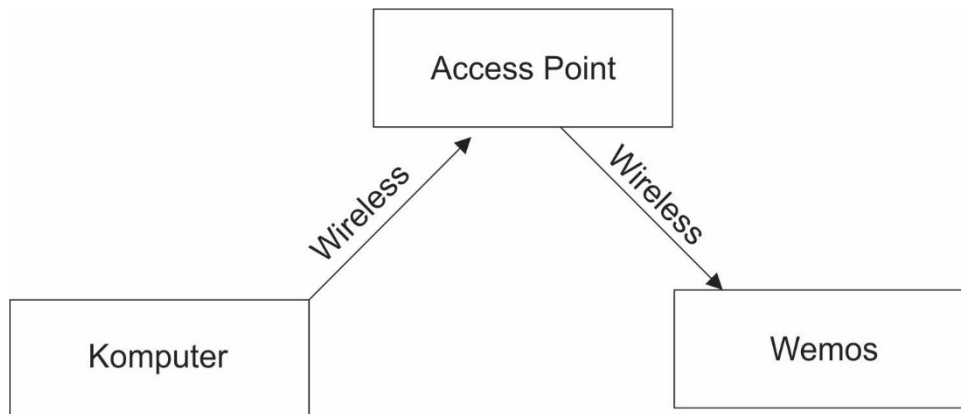
Jarak	Keterangan
Tanpa Penghalang	
3m	Terscaning
5m	Terscaning
10m	Terscaning
15m	Tidak terscaning
Dengan Penghalang	
2m	Terscaning
4m	Terscaning
6m	Terscaning
8m	Tidak terscaning

Dari hasil percobaan dapat disimpulkan jarak efektif Wemos dalam berkomunikasi maksimal tanpa penghalang adalah sejauh 10 meter dan dengan penghalang maksimal 6 meter. Dari data di atas jangkauan *range* *WiFi* Wemos dengan jarak efektif adalah 5 meter.

Hasil dari *scaning* di atas dapat dilihat pada Gambar 4.22

**Gambar 4.25** Hasil *scaning* *WiFi* (Kiri AP terlihat Kanan AP tidak terlihat)

#### 4.6 Pengujian Penerimaan Data Wemos Dari PC Melalui Telnet



**Gambar 4.26** Diagram blok pengujian penerimaan data dari PC menuju Wemos

Dalam pengujian ini dilakukan pengiriman data dari PC menuju ke Wemos menggunakan protokol telnet. Dari Gambar 4.23 dapat dilihat arah data adalah mulai dari komputer menuju ke *access point* lalu oleh *access point* akan dilanjutkan kepada Wemos.

Pada pengujian ini akan dikirimkan data sejumlah 1000 menuju ke Wemos sebanyak 10 kali dan data yang diterima Wemos akan dimonitor melalui *serial monitor* pada Arduino IDE serta dihitung waktu lama pengiriman data. Untuk pengujian pengiriman data ini juga akan digunakan parameter QoS sebagai acuan pengujian pelayanan jaringan.

##### 4.6.1 Tujuan

Pengujian ini dilakukan untuk memastikan Wemos dapat menerima data secara lengkap dan cepat serta tidak ada paket *loss* saat pengiriman, juga untuk mengetahui kualitas pelayanan jaringan.

#### 4.6.2 Alat yang digunakan

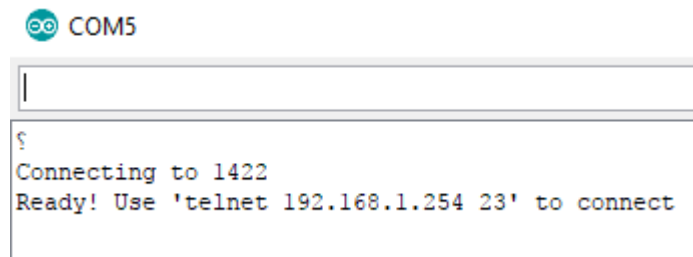
Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

- a. *Power supply 5V*
- b. *Akses Point* (Jaringan WLAN)
- c. *Software* Arduino IDE
- d. Laptop / Komputer
- e. Kabel data *mikro* USB
- f. Wemos D1 Mini
- g. *Software* pengirim data Wemos

#### 4.6.3 Prosedur pengujian

Prosedur pengujian :

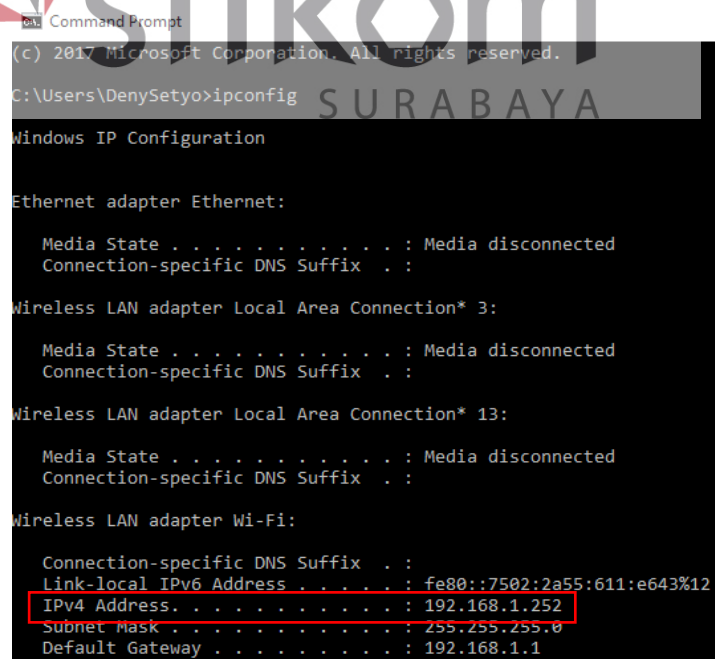
1. Nyalakan laptop atau komputer
2. Hubungkan Kabel data USB *mikro* pada Wemos D1 Mini yang akan menjadi *access point* dan sisi USB dihubungkan kepada USB laptop.
3. Buka *software* Arduino IDE.
4. Salin *Source code* dengan judul “Pengujian penerimaan data melalui telnet” pada lampiran
5. Pastikan *port* COM untuk komunikasi Wemos antara di Arduino IDE dan *device manager* sama.
6. *Upload* program yang telah dibuat menuju ke Wemos.
7. Setelah *upload* selesai buka *serial monitor*. Perhatikan Gambar 4.23.



**Gambar 4.27** Wemos siap untuk telnet

Dari Gambar 4.23 dapat disimpulkan bahwa Wemos sudah terhubung dengan *access point* dan sudah siap untuk menerima data. Dari gambar tersebut juga dapat diketahui bahwa alamat untuk Wemos berkomunikasi terdapat pada IP 192.168.1.254 dan *Port* 23.

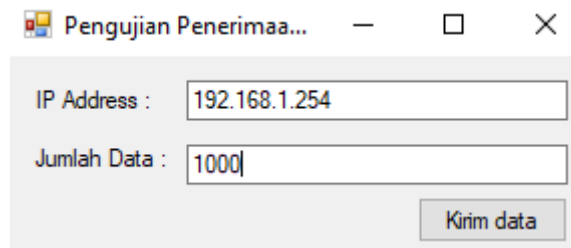
8. Sambungkan *WiFi* laptop kepada *access point* yang sama dengan Wemos yang akan menerima data
9. Buka *command prompt* dan lihat IP yang diterima oleh laptop berada pada 1 jaringan dengan Wemos penerima. Perhatikan Gambar 4.24.



**Gambar 4.28** IP yang di dapatkan laptop dari *acces point*

Dari Gambar 4.24 dapat diketahui bahwa antara laptop dan Wemos berada di 1 jaringan yang sama yaitu di jaringan 192.168.1.0.

10. Lakukan *ping* pada IP Wemos melalui *command prompt* bila Wemos membalas *ping* maka pengujian pengiriman data siap dimulai.
11. Buka *software* pengujian penerimaan data Wemos. Seperti pada Gambar 4.25

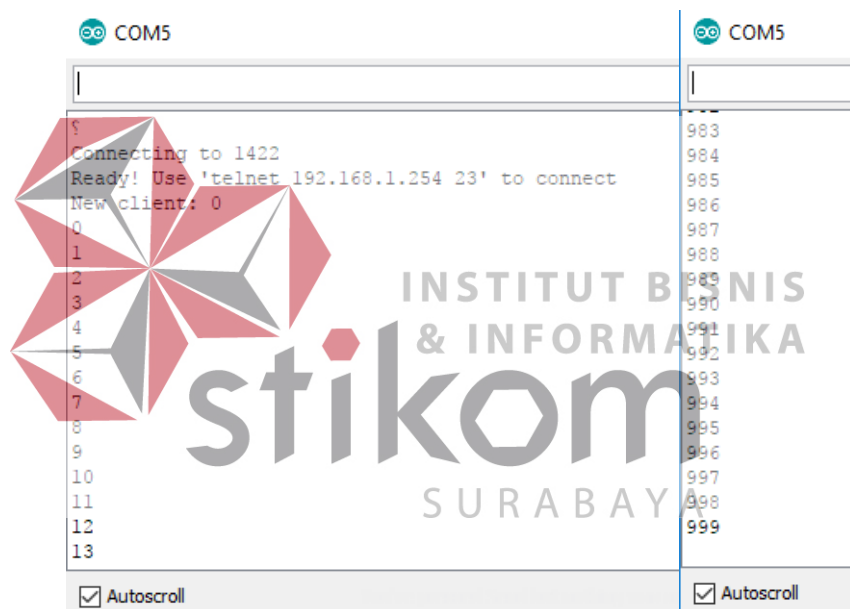


**Gambar 4.29** *Software* Pengujian penerimaan Wemos

Pada Gambar 4.25 adalah gambar tampilan *software* untuk pengujian penerimaan Wemos. Terdapat 2 kotak kolom yang harus diisi sebelum melakukan pengiriman data. Kolom pertama adalah IP Address ini berfungsi untuk menuliskan alamat IP Wemos yang akan dikirimkan data, dan pada kolom kedua adalah jumlah data dimana artinya diisi dengan banyak jumlah data yang akan dikirimkan oleh *software* ini menuju ke Wemos. Setelah semua terisi klik kirim data maka *software* ini akan mengirimkan data sebanyak jumlah data menuju ke IP address.

12. Setelah *software* mengirimkan data buka kembali *serial monitor* pada Arduino IDE. Perhatikan Gambar 4.26.
13. Pada pengujian digunakan juga pengukuran pelayanan dengan parameter *QoS* adapun pengukuran yang digunakan antara lain :

- 1) Pengukuran *delay* menggunakan rumus 2.1. *Delay* ini menunjukkan waktu tunda mulai dari user mengirimkan data sampai data tampil di *serial moitor* Wemos.
  - 2) Pengukuran *troughput* menggunakan rumus 2.2. *Troughput* ini menunjukkan kecepatan data pengiriman.
  - 3) Pengukuran paket *loss* menggunakan rumus 2.3. paket *loss* ini menunjukkan apakah data yang dikirim lengkap atau ada yang hilang
14. Pada pengujian ini pula



**Gambar 4.30** Data yang diterima Wemos (Kiri awal data kanan akhir data)

Dari Gambar 4.26 bahwa data yang diterima Wemos adalah mulai dari 0 – 999 artinya pengiriman yang dilakukan adalah berhasil karena Wemos menerima 1000 data tanpa *loss*.

#### 4.6.4 Hasil Pengujian

Dari Gambar 4.26 dapat dilihat bahwa saat pengiriman data tidak ada data yang *loss*. Hal ini menunjukkan bahwa hal ini dapat disimpulkan bahwa

pengiriman data Wemos 100% berhasil. Data dari 10 percobaan dapat dilihat pada Tabel 4.2.

**Tabel 4.2** Hasil 10 kali percobaan pengiriman data

Percobaan ke :	Data	Delay (ms)	Troughput(bps)	Packet Loss
1	1000	1,225	816,3	0
2	1000	1,230	813,01	0
3	1000	1,103	906,62	0
4	1000	1,112	899,28	0
5	1000	1,302	768,05	0
6	1000	0,979	1021,45	0
7	1000	0,981	1019,37	0
8	1000	1,024	976,56	0
9	1000	1,051	951,47	0
10	1000	0,946	1057,08	0
Rata - rata waktu		1,095	922,92	

Dari Tabel 4.2 dapat disimpulkan bahwa dari 10 percobaan pengiriman data kesepuluh percobaan berhasil karena tidak ada data *loss* bila dilihat dari tabel 2.4 bahwa bila nilai paket data 0 maka kondisi paket *loss* sangat baik. Bila mengacu pada tabel 2.3 dapat diketahui bahwa delay sangat bagus karena  $< 150$  ms. Untuk waktu yang ditempuh dalam pengiriman 1000 data memiliki waktu rata-rata 1095 ms atau bisa dikatakan 1 detik, juga rata-rata troughput yang digunakan adalah 922 bps data perdetik, dari hal ini dapat disimpulkan bahwa pengiriman data dapat dikatakan sangat cepat dikarenakan dalam mengirim 1000 data hanya memerlukan waktu 1 detik.

#### 4.7 Pengujian Sistem

Dalam pengujian sistem ini yang akan diuji adalah sistem pada Gambar 3.1. Adapun cara pengujianya akan dibuat 3 skenario pengujian antara lain :

Skenario 1 : Laptop akan mengirimkan data kepada Wemos sebanyak 30x lalu akan dilihat apakah terdapat kesalahan dalam percobaan.

Skenario 2 : Laptop akan mengirimkan data kepada Wemos 3 Wemos dan 5 LCD, mulai dari LCD kosong sampai terisi akan dilihat apakah ada kesalahan dalam percobaan tersebut.

Skenario 3 : Laptop akan mengirimkan data kepada 3 Wemos dan 5 LCD, mulai dari LCD berisi data sebelumnya sampai berubah ke data yang baru akan dilihat apakah ada kesalahan dalam percobaan tersebut.

#### 4.7.1 Tujuan

Pengujian sistem ini dilakukan untuk mengetahui apakah sistem dapat berjalan dengan lancar dalam menjalankan tugasnya dan untuk mengetahui apakah sistem ini mempunyai kesalahan dalam pelaksanaannya.

#### 4.7.2 Alat yang digunakan

Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

- a. 3 buah *Power supply* 5V
- b. *Software* Arduino IDE
- c. *Software* Product Price Display Wemos
- d. Laptop / Komputer
- e. 4 buah Kabel data *mikro* USB
- f. 3 buah Wemos D1 Mini
- g. 5 Buah LCD karakter 16x2
- h. 5 Buah Module I2C 1602



- i. Kabel LAN

#### 4.7.3 Prosedur pengujian

Prosedur pengujian :

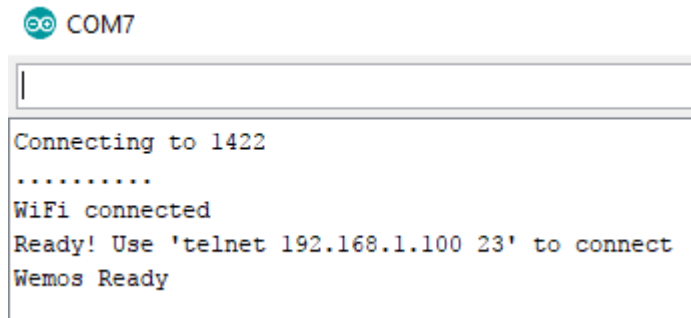
1. Nyalakan laptop atau komputer
2. Hubungkan Kabel data USB *mikro* pada Wemos D1 Mini ke 1.
3. Hubungkan Wemos ke 1 dengan modul LCD ke 1 menggunakan USB
4. Hubungkan modul LCD ke 1 dengan modul LCD ke 2 menggunakan USB
5. Buka *software* Arduino IDE.
6. Lakukan *scanning address* I2C untuk mendapatkan alamat I2C modul LCD.



Gambar 4.31 Hasil *Scanning* Wemos ke 1

Dari Gambar 4.27 dapat diketahui bahwa terdapat 2 modul LCD di Wemos ke 1. Dimana LCD ke 1 beralamat 0x3E dan LCD 2 beralamatkan 0x3F.

7. Salin *Source code* dengan judul “Pengujian sistem” pada lampiran
8. Pastikan *port* COM untuk komunikasi Wemos antara di Arduino IDE dan *device manager* sama.
9. *Upload* program yang telah dibuat menuju Wemos ke 1.
10. Setelah *upload* selesai buka *serial monitor*. Perhatikan Gambar 4.28.



**Gambar 4.32** IP address Wemos ke 1

Dari Gambar 4.28 menunjukkan Wemos sudah siap digunakan untuk pengiriman data melalui telnet dengan IP Wemos adalah 192.168.1.100.

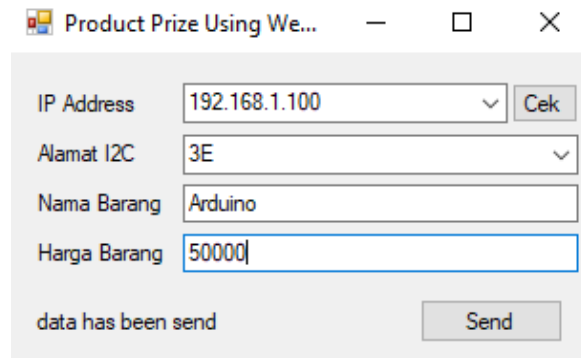
11. Lepas USB yang terhubung pada laptop pindah ke *power supply* 5V.
12. Perhatikan pada layar modul LCD bila di layar sudah tampil “LCD READY” maka LCD sudah siap untuk menampilkan data. Perhatikan Gambar 4.29.



**Gambar 4.33** Tampilan LCD menunjukkan LCD READY

Dari Gambar 4.29 diketahui bahwa modul LCD siap untuk digunakan dikarenakan pada tampilan LCD sudah menunjukkan LCD READY.

13. Buka *software* Product Price Display Wemos.
14. Setelah dibuka, isikan IP Wemos dan alamat I2C LCD ke 1. Perhatikan Gambar 4.30.



Product Prize Using We...

IP Address: 192.168.1.100 [Cek]

Alamat I2C: 3E

Nama Barang: Arduino

Harga Barang: 50000

data has been send [Send]

**Gambar 4.34** Tampilan *software Product Price Display Using Wemos*

Dari Gambar 4.30 diketahui bahwa kiriman ditujukan kepada Wemos ke 1 karena alamat IP menunjukan Wemos ke 1. Dari alamat I2C menunjukkan data akan dikirimkan kepada LCD ke 1 karena alamat I2C menunjuk ke alamat LCD ke 1. Lalu yang akan dikirimkan adalah data berupa nama barang yaitu Arduino dan harga barang adalah 50000 ini menunjukkan LCD ke 1 digunakan untuk *display* Arduino yang memiliki harga Rp. 50000. Setelah selesai data benar lalu klik *send* maka data akan dikirimkan pada Wemos ke 1.

15. Data yang dikirimkan diterima oleh Wemos ke 1 dan ditampilkan pada LCD ke 1. Perhatikan Gambar 4.31.



**Gambar 4.35** Data berhasil di tampilkan oleh LCD ke 1

Dari Gambar 4.31 dapat dilihat bahwa LCD ke 1 menampilkan data yang dikirimkan melalui *software* Product Price Display Wemos. Dari gambar tersebut juga dapat disimpulkan bahwa pengiriman data berhasil.

16. Lakukan perubahan alamat I2C menjadi 3F untuk melakukan pengiriman data pada Wemos ke 1 LCD ke 2. Lalu pada nama barang isikan Wemos dan pada harga barang isikan 10000. Lalu perhatikan apakah LCD ke 2 menampilkan data nama barang dan harga barang yang sesuai dengan yang dikirimkan. Perhatikan gambar 4.32.



Gambar 4.36 Data berhasil ditampilkan oleh LCD ke 2

17. Hubungkan Wemos ke 2 dengan modul LCD ke 3 menggunakan USB
18. Hubungkan modul LCD ke 3 dengan modul LCD ke 4 menggunakan USB
19. Buka *software* Arduino IDE.
20. Lakukan *scanning address* I2C untuk mendapatkan alamat I2C modul LCD.
21. *Upload* program menuju Wemos ke 2.
22. Setelah *upload* selesai buka *serial monitor*. Pastikan Wemos kedua mendapat IP yang berbeda dengan IP pertama
23. Lepas USB yang terhubung pada laptop pindah ke *power supply* 5V.
24. Lakukan langkah 17 – 23 untuk Wemos ke 3 dan LCD 5

25. Setelah semua Wemos dan LCD dapat digunakan atau dalam kondisi *ready* lakukan skenario 3 percobaan.

#### 4.7.4 Hasil Pengujian

Dari 3 skenario percobaan didapatkan hasil dari skenario seperti dapat dilihat pada Tabel 4.3.

**Tabel 4.3** Hasil percobaan skenario

No.	Data Dikirim		Tujuan		Keterangan
	Nama barang	Harga Barang	Wemos ke -	LCD ke -	
1	Arduino	Rp 50.000	1	1	Berhasil Terkirim
2	Wemos	Rp 10.000	1	2	Berhasil Terkirim
3	Resistor	Rp 100	2	3	Berhasil Terkirim
4	Capasitor	Rp 250	2	4	Berhasil Terkirim
5	Smartphone	Rp 2.750.000	3	5	Berhasil Terkirim
6	Sarimi Isi 2	Rp 6.700	1	1	Berhasil Terkirim
7	Indomie goreng	Rp 3.200	1	2	Berhasil Terkirim
8	Forvita 200g	Rp 9.900	2	3	Berhasil Terkirim
9	Bakmi Mewah	Rp 9.900	2	4	Berhasil Terkirim
10	Sunkara	Rp 7.900	3	5	Berhasil Terkirim
11	Ayam Brand	Rp 20.900	1	1	Berhasil Terkirim
12	Prochiz	Rp 14.900	1	2	Berhasil Terkirim
13	SGMEx 1 900g	Rp 69.200	2	3	Berhasil Terkirim
14	SGMEx 3 400g	Rp 32.000	2	4	Berhasil Terkirim
15	SGMEx 3 900g	Rp 31.250	3	5	Berhasil Terkirim
16	PROMINA Bubur	Rp 11.150	1	1	Berhasil Terkirim
17	Bebelac Gold 3	Rp 114.500	1	2	Berhasil Terkirim
18	SUN Bubur	Rp 6.000	2	3	Berhasil Terkirim
19	Nutrilon 3	Rp 12.500	2	4	Berhasil Terkirim
20	Morin Jam Choco	Rp 18.600	3	5	Berhasil Terkirim
21	Tropicana Slim	Rp 37.400	1	1	Berhasil Terkirim
22	Heavenly Blush	Rp 5.900	1	2	Berhasil Terkirim
23	WRP On The Go	Rp 9.900	2	3	Berhasil Terkirim
24	Kaki Tiga	Rp 3.000	2	4	Berhasil Terkirim
25	Aqua 1500ml	Rp 4.700	3	5	Berhasil Terkirim
26	Aqua Click 750ml	Rp 5.000	1	1	Berhasil Terkirim
27	Belvita 80g	Rp 5.900	1	2	Berhasil Terkirim
28	Monde Serena	Rp 16.900	2	3	Berhasil Terkirim
29	Tao kae noi	Rp 8.900	2	4	Berhasil Terkirim
30	Kinder joy	Rp 9.900	3	5	Berhasil Terkirim

Dari tabel percobaan di atas dapat memenuhi 3 skenario yang diujikan. Dapat dipisah sebagai berikut :

1. Pada saat pengiriman nomor 1 -5 didapati bahwa ini sesuai dengan skenario kedua dimana *user* harus mengisi data mulai dari LCD kosong atau *ready* menjadi menampilkan data yang dikirimkan.

Adapun hasil dari pengujian tersebut adalah seperti ditunjukkan oleh Tabel 4.3. Data berhasil dikirimkan ke alamat yang tepat.

2. Pada saat pengiriman nomor 6 -10 didapati bahwa ini sesuai dengan skenario ketiga dimana *user* harus mengganti data yang ditampilkan LCD dari data lama menjadi data baru.

Adapun hasil dari pengujian tersebut adalah seperti ditunjukkan oleh Tabel 4.3. Data berhasil dikirimkan dan tampilan LCD berubah sesuai data baru yang dikirimkan.

3. Pada saat pengiriman nomor 1 - 30 didapati bahwa ini sesuai dengan skenario pertama dimana *user* harus mengirimkan data sebanyak 30 data secara terus - menerus.

Adapun hasil dari pengujian tersebut adalah seperti ditunjukkan oleh Tabel 4.3. Data berhasil dikirimkan semua dan tidak ada data hilang ataupun salah alamat.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Adapun kesimpulan yang dapat dituliskan setelah melakukan analisa dan percobaan dari sistem yang diteliti antara lain sebagai berikut :

1. Sistem *Product Price Display Using Wemos* ini dapat digunakan sebagai alternatif teknologi untuk label harga dikarenakan pada saat pengujian tidak pernah ada data yang tidak terkirim, semua data yang dikirim membutuhkan waktu kurang dari 1 detik.
2. *Software* yang digunakan dalam sistem ini cukup mudah, pada tampilan *software* dapat dilihat pada Gambar 4.30. Dimana *user* cukup mengetik alamat yang dituju dan data yang akan dikirimkan lalu klik kirim data maka label harga di rak barang akan langsung berubah sesuai dengan data yang dikirimkan.
3. Dari *design* sistem ini juga dapat dikatakan *portable* karena *LCD display* dapat dilepas dan diganti pada setiap modulnya dengan mudah. Serta untuk sambungannya sendiri hanya cukup menggunakan kabel USB *male to male* yang langsung ditancapkan.
4. *Product Price Display Using Wemos* dapat memudahkan swalayan dalam melakukan operasionalnya, juga dapat membuat sebuah swalayan lebih modern dan canggih.

5. Pada pelaksanaannya sistem ini akan dapat menjadi alternatif sebagai *display* harga pada rak barang karena sistem sudah tersentral dan ketika harga di kasir dirubah maka *display* di rak juga akan berubah.

## 5.2 Saran

1. *Product Price Display Using Wemos* perlu dikembangkan dalam sisi tampilan dimana tampilan yang dimunculkan belum dapat memenuhi kriteria label harga yang diatur oleh kementerian perdagangan, dimana label harga harus disertai barcode.
2. Tampilan untuk *Product Price Display Using Wemos* ini masih sangat terbatas yaitu hanya 16 karakter untuk nama barang. Pada kenyataanya nama barang yang dijual di pasaran memiliki nama barang yang cukup panjang dan lebih dari 16 karakter.
3. Perlu ditambahkan fitur auto *scanning* pada I2C agar ketika modul LCD diganti tidak perlu melakukan registrasi ulang terutama melalui program.
4. Pada sisi *software* sistem ini membutuhkan sebuah *software inventory* yang lebih memudahkan dalam pengoperasiannya.



## DAFTAR PUSTAKA

Agung, Rizky. 2013. Tutorial Cara Setting Mikrotik RB750 via Winbox. [Online], (<https://mikrotikindo.blogspot.co.id/2013/04/tutorial-cara-setting-mikrotik-rb750-winbox.html>, diakses 25 Januari 2017)

Arafat, R.H.. 2016. *Sistem Pengamanan Pintu Rumah Berbasis Internet Of Thinks (IoT) Dengan ESP8266*. "Technologia" Vol 7, No.4, Oktober – Desember 2016. Dambil dari : <http://ojs.uniska.ac.id/index.php/JIT/article/download/661/578> , pada tanggal 1 Juli 2017

Ardan, Dani. 2016. Panduan lengkap penggunaan 1602 / 2004 I2C LCD untuk Arduino. [online], ( <http://www.belajarduino.com/2016/06/how-to-connect-1602-2004-iic-i2c-lcd-to.html>, diakses 25 Januari 2017)

Basu Swastha, dan Irawan. 2003. *Manajemen Pemasaran Modern*. Yogyakarta, : Liberty Yogyakarta

Kadir, Abdul. 2003. *Jaringan Komputer*. Yogyakarta : Andi

pccontrol.wordpress.com. 2011. Pengetahuan Dasar pemrograman C untuk I2C AVR dgn codevision. [online], (<https://pccontrol.wordpress.com/2011/06/26/pengetahuan-dasar-pemrograman-c-untuk-i2c-avr-dgn-codevision.html>, diakses 25 Januari 2017)